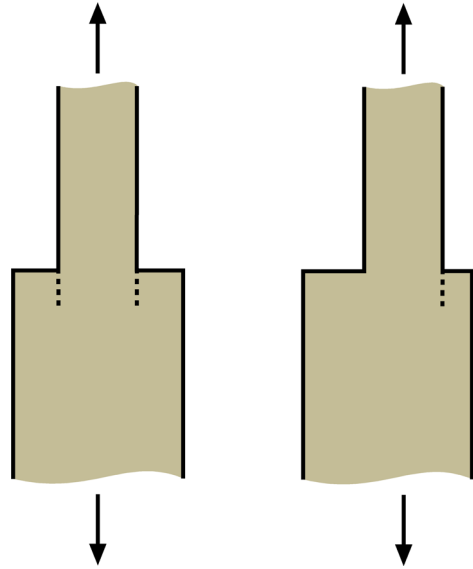




LUND
UNIVERSITY



SPRICKLAST FÖR DRAGBELASTAT TRÄ MED SEKTIONSÄNDRING

ERIK GOTTSÄTER och DAVID KINSELLA

Structural
Mechanics

Master's Dissertation

DEPARTMENT OF CONSTRUCTION SCIENCES
DIVISION OF STRUCTURAL MECHANICS

ISRN LUTVDG/TVSM--14/5191--SE (1-161) | ISSN 0281-6679

MASTER'S DISSERTATION

SPRICKLAST FÖR DRAGBELASTAT TRÄ MED SEKTIONSÄNDRING

ERIK GOTTSÄTER och DAVID KINSELLA

Supervisors: Professor **PER JOHAN GUSTAFSSON** and **HENRIK DANIELSSON**, PhD;
Div. of Structural Mechanics, LTH, Lund.

Examiner: Professor **ROBERTO CROCETTI**, Div. of Structural Engineering, LTH, Lund.

Copyright © 2014 Division of Structural Mechanics
Faculty of Engineering (LTH), Lund University, Sweden.
Printed by Media-Tryck LU, Lund, Sweden, May 2014 (*Pl*).

For information, address:
Div. of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: <http://www.byggmek.lth.se>

Förord

Denna rapport skrevs som examensarbete vid Lunds Tekniska Högskola, Avdelningen för byggnadsmekanik, i samarbete med Limträteknik AB i Falun. Uppkomsten av problemformuleringen till arbetet härstammar från diskussioner med Prof. Per Johan Gustafsson och Arne Emilsson på Limträteknik AB.

Vi skulle vilja tacka Roberto Crocetti på Avdelningen för konstruktions-teknik, LTH, för värdefulla upplysningar och personalen på byggnadsmekanik, LTH, bland andra Kent Persson som bistått med goda råd om brottmekanik och modellering med ABAQUS. Tack också Zivorad Zivkovic på Avdelningen för hållfasthetslära, LTH, för att vi fick tillgång till dragmaskinen i M-huset och kunde genomföra provningen. Särskilt vill vi tacka våra handledare Prof. Per Johan Gustafsson och TeknD. Henrik Danielsson för att de alltid varit så beredda att diskutera frågor och generöst delat med sig av sin tid. Vi har känt oss inspirerade och motiverade av deras bemötande och haft ett gott stöd hela vägen från början till slut.

Lund, 22 maj 2014

David Kinsella

Erik Gottsäter

Abstract

On occasion, structural wooden members in tension feature a sectional alteration in the form of a notch. In spite of being a highly inhomogeneous and anisotropic material, as an approximation wood may be modelled as homogeneous and orthotropic. By means of such a material model, first order linear elastic theory and a fracture mechanics condition on the strain energy release rate, a formula is derived for the load bearing capacity with respect to fracturing. The capacity is further simulated in 2D with FEM both by employing a linear elastic method with node releasing and by a non-linear method with a cohesive zone. The validity of the analytical and numerical models are verified by laboratory tests comprising 21 small wooden members with varying notch size. The characteristic value of the capacity is determined by the 5th percentile with 75 % confidence level. The results show that the formula is successful in predicting the load bearing capacity of the wooden members which were tested. However, the size effect has yet to be established.

Sammanfattning

Ibland görs sektionsändring i form av rektangulära urtag i dragbelastade konstruktionsdelar av trä. Även om trä är ett inhomogent och anisotrop material, så kan det approximativt modelleras som homogent och ortotrop. Ett uttryck för bärförmågan med avseende på uppsprickning härleds med utgångspunkt i 1) en förenklad materialmodell med antagande om homogenitet och otropi, 2) första ordningens linjärelastiska teori och 3) ett brottmekaniskt villkor på den töjningsenergi som frigörs vid uppsprickning. Bärförmågan simuleras också med 2D finita element dels med en linjärelastisk metod som bygger på att släppa upp noder och dels med en olinjär metod som bygger på en kohesiv zon. De analytiska och numeriska resultaten verifieras genom provning av 21 små träkroppar med varierande urtagsdjup. Det karakteristiska värdet på bärförmågan härleds som 5-percentilen med 75 % konfidensgrad. Resultaten visar att beräkningsuttrycket med fördel kan användas för att beskriva bärförmågan hos de små träkropparna som provades. Emellertid har storlekseffekten inte verifierats. Resultaten visar också att de båda numeriska beräkningsmodellerna är nära nog lika bra på att förutsäga bärförmågan hos de provade träkropparna.

Nyckelord: sektionsändring, urtag, trä, brottmekanik, noduppsläppning, kohesiv zon, verifiering genom provning

Populärvetenskaplig sammanfattning

Trä är ett viktigt byggnadsmaterial med en betydande industri i Sverige. Exempel på träkonstruktionsdelar är dragstag, strävor, balkar och ramar. För att kunna passa ihop och förbinda flera delar i en knutpunkt görs ibland sektionvändning i form av rektangulära urtag. När en sådan konstruktionsdel dragbelastas uppstår spänningskoncentrationer vid urtagets hörn. Dessa kan leda till sprickbildning och brott. Bärförmågans storlek med hänsyn till uppsprickning utreds genom härledning av ett beräkningsuttryck.

Härledningen av beräkningsuttrycket bygger på en förenklad materialmodell för trä, elasticitetsteori och brottmekanik. Genom att utgå från den energi som omsätts när en spricka tillväxer kan bärförmågan uttryckas som funktion av bland annat tvärsnittshöjden, urtagsdjupet och den materialspecifika energiåtgången vid uppsprickning. Verknings sättet och bärförmågan simuleras med två datormodeller. Den ena bygger på modellantaganden liknande dem i härledningen av beräkningsuttrycket. Den andra bygger på mer raffinerade antaganden som förväntas ge en bättre respons. Beräkningsuttrycket och de båda datormodellerna verifieras genom provning av 21 stycken små och nära nog "felfria" träkroppar med varierande urtagsdjup. Med felfri menas här att förekomsten av kvistar, sprickor och andra naturliga inslag är nära nog obefintlig.

Resultaten visar att beräkningsuttrycket med god noggrannhet kan användas för att beskriva uppsprickningslasten hos de provade träkropparna och att båda datormodellerna är nära nog lika goda på att beskriva verknings sättet. Resultaten visar också att beräkningsuttrycket i princip kan användas för att beskriva uppsprickningslasten för lastfall som inte direkt omfattas av de aktuella provningarna. Emellertid behöver fler provningar genomföras på stora träkroppar för att verifiera storlekseffekten.

Innehåll

Förord	i
Abstract	iii
Sammanfattning	v
Populärvetenskaplig sammanfattning	vii
1 Inledning	1
1.1 Syfte och avgränsning	2
2 Bakgrund och teori	7
2.1 Exempel på verkliga urtag	8
2.2 Materialet trä	11
2.2.1 Inledning	11
2.2.2 Bildnings sätt och egenskaper	11
2.2.3 Europeiska normer	14
2.2.4 Materialmodellering	16
2.3 Brottmekanik	17
2.3.1 Inledande definitioner	17
2.3.2 Spänningstillståndet vid sprickspetsen med antagande om linjärelasticitet	17
2.3.3 Principen om energijäm vikt	20
2.3.4 Kohesiv zon	23
2.3.5 Så uppmäts brottenergi	27
2.3.6 Ökända sprickor i modern tid	27
2.3.7 Sprickor i populärkulturen	28
2.4 ABAQUS	31
2.4.1 Allmänt	31
2.4.2 Att implementera Python-skript i ABAQUS	32

3	Försöksplanering	37
3.1	Allmänt	38
3.2	Dragmaskinen	38
3.3	Dimensionering av provkroppar	39
3.3.1	Dimensionering med hänsyn till lastspredningseffekten	39
3.3.2	Dimensionering med hänsyn till infästningsanordningen	41
3.4	Utförande av provkroppar	42
3.5	Genomförande av provning	43
4	Modellantaganden	47
4.1	Beräkningsmodeller	48
4.1.1	Analytisk metod för bestämning av uppsprickningslast	48
4.1.2	FEM med noduppsläppning	54
4.1.3	FEM med kohesiv zonmodellering	58
4.2	Slumpmodell	62
5	Metod	65
5.1	Verifiering och kalibrering genom provning och simulering .	66
5.1.1	Verifiering av beräkningsformel för två likadana urtag med centrisk belastning	66
5.1.2	Kalibrering av numeriska beräkningsmodeller genom provning	67
5.1.3	Bestämning av den skenbara brottsegheten \tilde{K}_c genom simulering	68
5.1.4	Verifiering av beräkningsformel allmänna fallet . . .	70
5.2	Statistiska metoder	70
5.2.1	Konfidensintervall för normalpercentiler	70
5.2.2	Simultana konfidensintervall för tre normala punktskattningar	72
5.2.3	Kolmogorov-Smirnovs test	73
6	Resultat	75
6.1	Verifiering och kalibrering genom provning och simulering .	76
6.1.1	Provningsresultat	76
6.1.2	Verifiering av beräkningsformel för två likadana urtag med centrisk belastning	77
6.1.3	Kalibrering av numeriska beräkningsmodeller genom provning	80

6.1.4 Bestämning av den skenbara brottsegheten \widetilde{K}_c genom simulering	81
6.1.5 Modblandningen i några speciella fall	81
6.1.6 Verifiering av beräkningsformel allmänna fallet	82
7 Slutsatser	97
7.1 Diskussion	98
7.2 Beräkningsuttryck i konventionella storheter	100
7.3 Förslag på vidare forskning	106
7.4 Sammanfattning	106
Källor	109
Bilaga 1	115
Bilaga 2	127
Bilaga 3	133
Bilaga 4	147
Bilaga 5	159

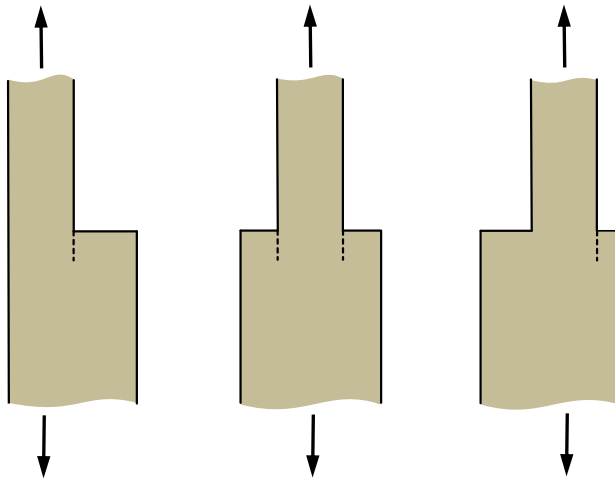
1 Inledning

1.1 Syfte och avgränsning

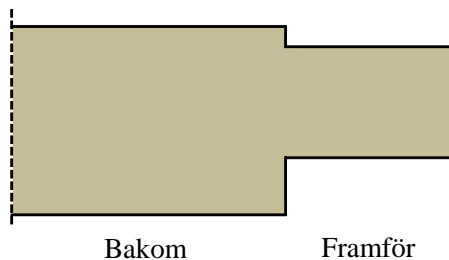
I dragbelastade konstruktionsdelar av trä finns ibland sektionssändring av den typ som visas i Figur 1.1. Sektionsändringen åstadkoms med urtag och avsikten är oftast att olika stora träkroppar ska kunna passas ihop och förbindas i en knutpunkt. Vid urtagshörnet riskerar en spricka att bildas som kan växa och leda till brott. Det saknas emellertid en beräkningsmetod som konstruktören kan använda för att bestämma den last som ger sprickbildning.

Syftet är att härleda ett beräkningsuttryck för bestämning av bärförmågan med avseende på uppsprickning vid urtag. Beräkningsuttrycket verifieras genom provning och simulering och karakteristiskt värde bestäms. Frågeställningarna är

- Hur bestäms bärförmågan med avseende på uppsprickning i en dragbelastad träkonstruktionsdel med sektionssändring med ett praktiskt beräkningsuttryck?
- Hur verifieras en beräkningsformel genom provning och datorsimulering?
- Hur bestäms karakteristiskt värde av bärförmågan uttryckt i konventionella storheter?



Figur 1.1: Exempel på dragbelastade delar med sektionssändring i form av rektangulära urtag.

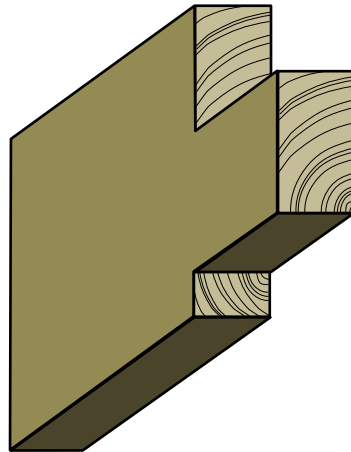


Figur 1.2: Figuren illustrerar vad som menas med “framför” respektive “bakom” urtaget.

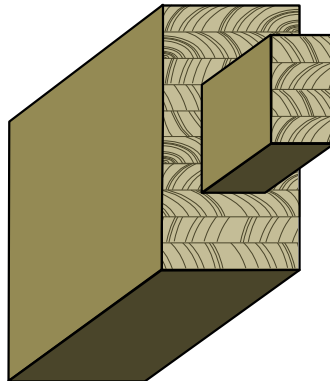
Avgränsningarna är

- Sektionsändringens geometri kan beskrivas i planet såsom visas i Figur 1.1 och Figur 1.3a. Med andra ord beaktas inte den sortens urtag vars geometri måste beskrivas i rummet. Ett exempel på urtag som således inte beaktas ges i Figur 1.3b.
- Sektionsändringen består av ett eller två rektangulära urtag belägna vid kroppens ände och vid tvärsnittets ytterkant(er). När två urtag görs är de lika långa i kroppens längsriktning.
- Provning görs endast på kroppar med två likadana urtag och med centrisk normalkraft.
- Provning avser endast små provkroppar av “felfritt” trä.
- Den enda parameter som varierar i provningen är urtagsdjupet och den varierar på tre nivåer.
- I simuleringarna modelleras materialet som tvådimensionellt, homogent och ortogonalt anisotropt.
- Effekten av lastens varaktighet utreds inte.

Vad som menas med “framför” respektive “bakom” urtaget framgår av Figur 1.2. Examensarbetet är disponerat enligt schemat i Figur 1.4.

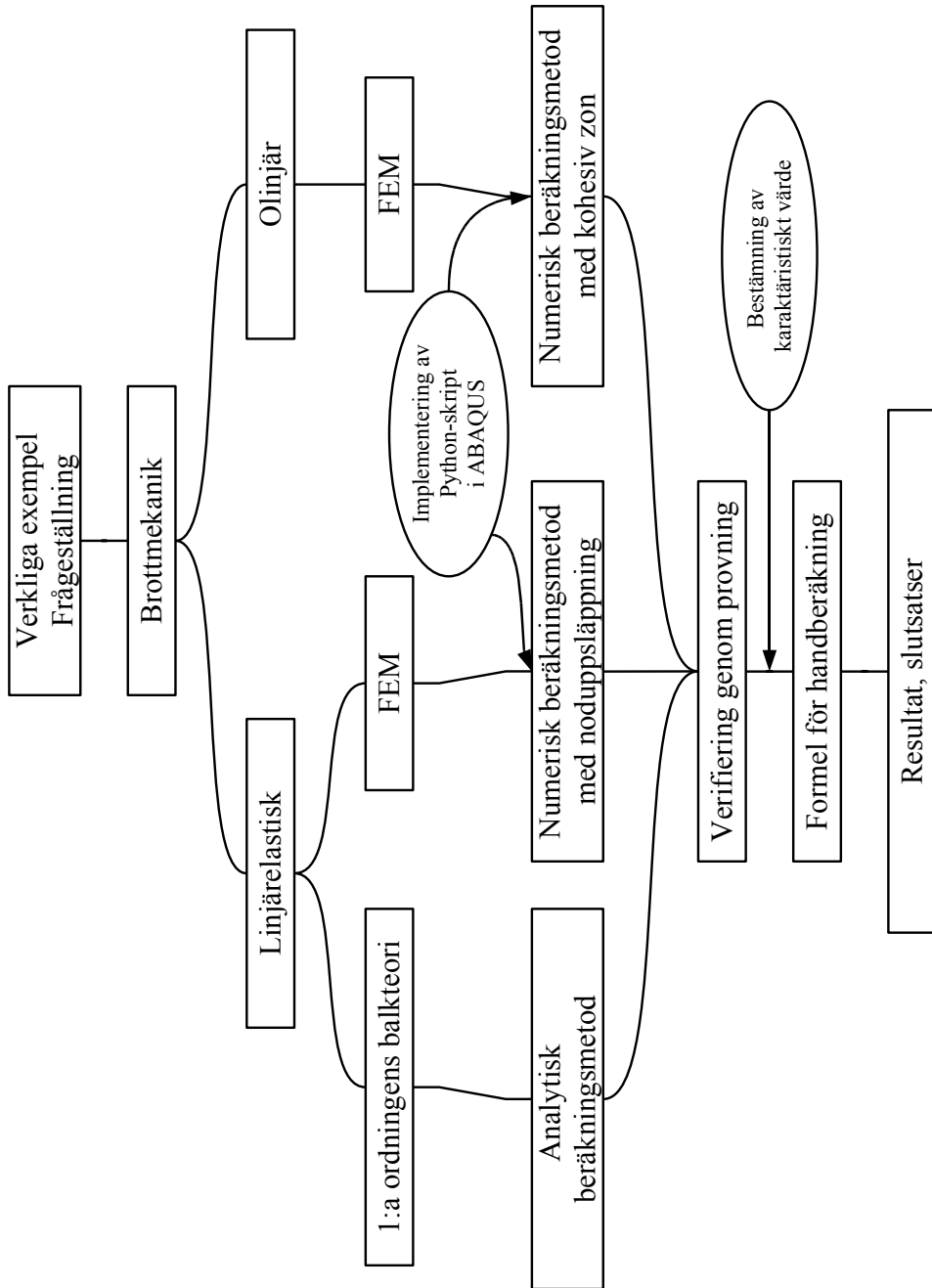


(a)



(b)

Figur 1.3: a) visar ett urtag som geometriskt sett kan beskrivas i planet. Denna sortens urtag studeras i föreliggande examensarbete. b) visar ett urtag som geometriskt sett kräver en beskrivning i rummet. Denna sortens urtag behandlas inte i examensarbetet.



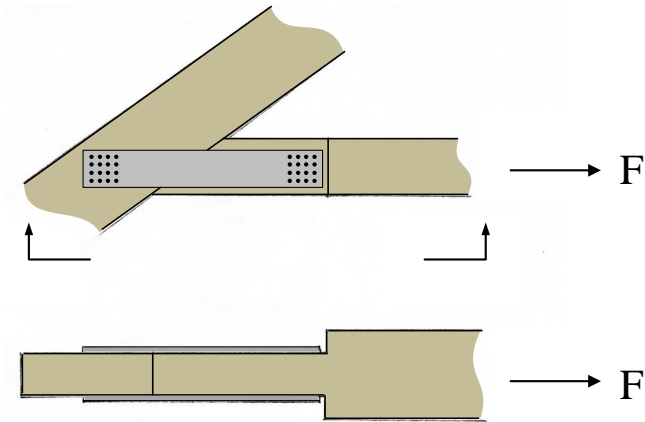
Figur 1.4: Examensarbetet är disponerat enligt schemat.

2 Bakgrund och teori

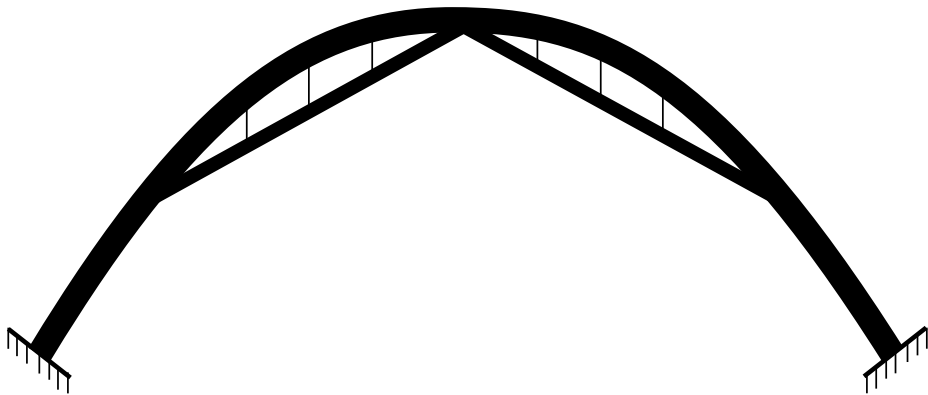
2.1 Exempel på verkliga urtag

Tvåsidiga urtag görs oftast så att bredden på konstruktionsdelen ska passa med bredden på en eller flera andra konstruktionsdelar i ett förband av till exempel spikplåt eller dymling. Ett exempel på detta ges i Figur 2.1 som visar en detaljskiss inspirerad av en verklig konstruktion i norra Europa. Denna typ av detalj kan påträffas i en bågkonstruktion med strävor under bågen såsom illustreras Figur 2.2. Strävans tvärsnittshöjd har i tre fall reducerats från 280 mm till 215 mm respektive från 140 mm till 90 mm samt från 330 mm till 215 mm. I samtliga fall gäller att båda urtagen är lika djupa på vardera sidan längsaxeln. Anledningen till att strävan är bredare mellan ändarna är att detta minskar knäckningsrisken, vilket ofta är dimensionerande för denna typ av konstruktionsdel. Knäckning kan uppkomma då strävan är tryckbelastad. Men en sträva av denna typ skulle lika gärna kunna bli dragbelastad. Därför är det mycket relevant att utreda dragbelastade konstruktionsdelar med urtag.

Ensidiga urtag förekommer främst vid upplaget av böj- och tvärkraftsbelastade balkar. I SS-EN 1995-1-1 [15] finns anvisningar för dimensionering av denna typ av konstruktionsdel med hänsyn till tvärkraft. Dragbelastade delar med ensidiga urtag går dock att hitta i vissa konstruktioner. Ett exempel ges i Figur 2.3 som föreställer dragkrysset i en pergola i Folkets park, Lund, som uppfördes 1976 [27]. Pergolan löper hela vägen längs parkbyggnadens fasad mot sydväst. Trästrävorna befinner sig i samma plan och möts i mitten. Varje strävas tvärsnitt halveras för att båda strävorna ska vara kontinuerliga. Intill urtagen syns en ca 15 cm lång spricka, se Figur 2.4. Uppsprickningen kan ha orsakats av dragbelastning när pergolan utsatts för vindlast från nordost eller sydväst. Exemplet visar på sårbarheten i denna typ av konstruktion.



Figur 2.1: Exempel på konstruktionsdel med urtag vid infästning.



Figur 2.2: Skiss över bärande stomme med två strävor där urtag kan bli aktuella att använda.



Figur 2.3: Dragkryss av trä i Folkets park i Lund. Dragkryssets strävor befinner sig i samma plan, därför görs urtag i mitten på strävorna.



Figur 2.4: På de nedre skuggade sidorna av strävorna kan sprickor urskiljas vilka begynner vid urtagens hörn.

2.2 Materialet trä

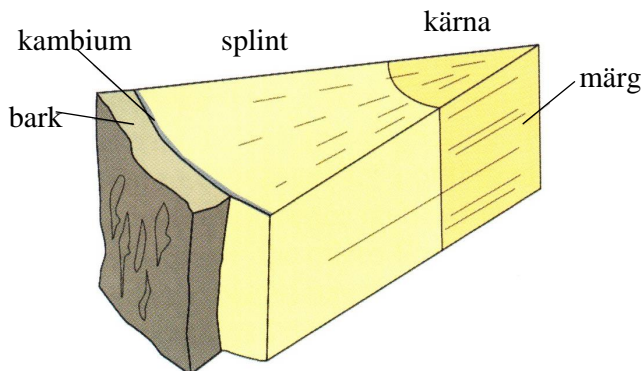
2.2.1 Inledning

Trä är ett av de allra äldsta byggnadsmaterialen. Det är till exempel lätt att föreställa sig hur en stock med enkla medel kunde tjäna som bro i förhistorisk tid. Willy Tegel [51] omtalar fyra vattenbrunnar i nuvarande östra Tyskland tillverkade i trä och ca 7000 år gamla som bevis för att neolitiska jordbrukare var kunniga snickare. Med tiden har en betydande skogsindustri vuxit fram som år 2013 omfattar 10-12 % av den svenska industrins totala omsättning och sysselsättning. Den svarar också för cirka 12 % av total svensk export till ett värde av 124 miljarder kronor [48]. Det är kombinationen av låg egenvikt, hög hållfasthet, god isoleringsförmåga och beständighet som gör trä så användbart, förutom de estetiskt tilltalande egenskaperna. I Sverige är gran (*pinus abies*) och furu (*pinus silvestris*) de vanligaste träslagen för konstruktionsändamål. De båda träslagen skiljer sig åt endast försumbart med avseende på sina byggnadstekniska egenskaper förutom att gran inte kan tryckimpregneras [24].

2.2.2 Bildningsätt och egenskaper

En trädstam består i genomskärning av bark, kambium, splint, kärna och märg och en illustration av detta finns i Figur 2.5. I kärnan av gran och furu produceras hartser. Dessa fungerar som kemiskt skydd mot biologiska angrepp. Det är endast i kambiet, ett tunt lager celler, som ny ved produceras. Om våren och sommaren varje år tillväxer trädet, varvid nya årsringar bildas. Vårveden har fibrer med tunnare och glesare cellväggar och lägre densitet än sommarveden som dessutom brukar ha ett mörkare utseende [24]. Trämaterialet består på mikroskopisk skala av ihåliga fibrer, cirka 1-2 mm långa och uppemot 0,1 mm i diameter [48]. En fiber består i huvudsak av cellulosa. Fibrerna är organiserade i buntar som hålls ihop av lignin, ett naturligt lim.

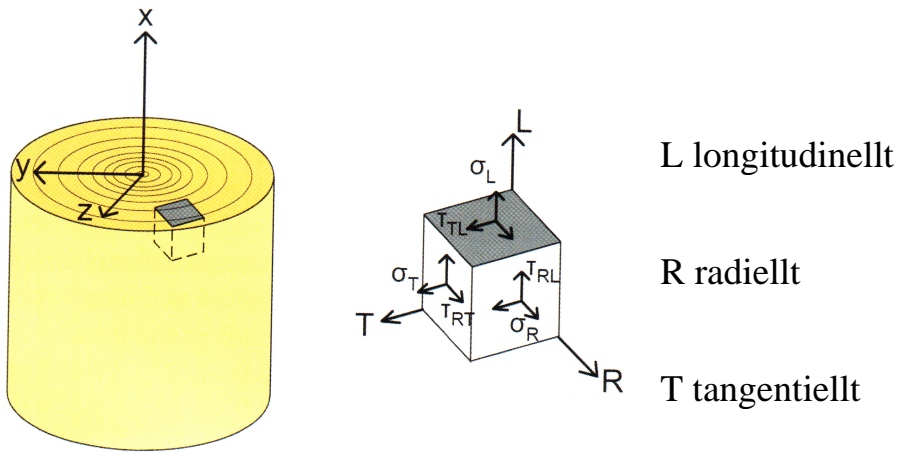
Trä är anisotropt vilket betyder att det har olika egenskaper i olika riktningar [11]. Längs med fibrerna har träet en relativt hög hållfasthet i både tryck och drag medan draghållfastheten är endast 5-10 % så stor tvärs fiberriktningen [24]. Dessutom skiljer sig egenskaperna åt tvärs fiberriktningen, beroende på om materialet betraktas parallellt med årsringarna eller vinkelrätt dessa. För ett litet trästycke utan defekter



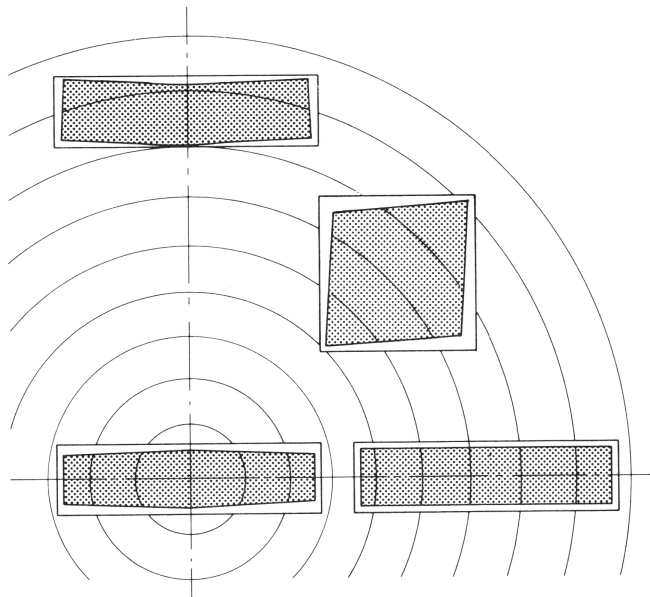
Figur 2.5: Trädstam i genomskärning. Bildkälla: [48].

kan det med god noggrannhet antas att träet har ortogonalt anisotropa (ortotropa) egenskaper [11]. Huvudriktningarna är illustrerade i Figur 2.6. Hållfastheten för trä varierar ungefär linjärt med densiteten [24]. I litteraturen anges normalt densiteten vid en fuktkvot på 12 %. Fuktkvoten definieras som förhållandet mellan fukttinnehåll och torrdensitet. Trä är ett hygroskopiskt material och fuktförhållandena har stor betydelse för trädets verkningssätt [3]. Trädets egenskaper varierar med lastvaraktigheten och temperaturen [41, 24]. Trä är också ett viskoelastiskt material vilket betyder att det påverkas av långtidseffekter såsom krypningsfenomen [3].

Ett träd är optimerat för att överleva i sin naturliga miljö. Några bevis på detta är stammens cylindriska form som minimerar effekten av vindlast, fibrernas struktur och uppbyggnad som optimerar hållfastheten mot vindbelastning och egentynghd, kvistar som förser trädet med blad och barr vilka möjliggör fotosyntes, transpiration och respiration samt hartser som tillhandahåller ett naturligt kemiskt skydd mot eventuella biologiska angrepp. Flera av de egenheter som gör trädet optimalt för sin naturliga miljö kommer av en ingenjör att betraktas som defekter i ett byggnadsmaterial. Vanliga så kallade “fel” är kvistar, sprickor, snedfibrighet, tjurved och kådlåpor [25]. Trädet kan dessutom angripas och nedbrytas av insekter och röta. I ett ekologiskt perspektiv är detta en stor fördel på grund av att trä ingår i ett naturligt kretslopp [24]. När trädet sågas och torkas kan det skevas och kupas beroende på hur fibrerna och årsringarna i trästycket varit orienterade från början. I Figur 2.7 illustreras hur ett trästycke kan tänkas skevas och kupas



Figur 2.6: Ett litet homogent och felfritt trästycke kan betraktas som ett ortotrop material med huvudriktningar enligt figuren. Bildkälla: [48].



Figur 2.7: Årsringarnas orientering påverkar hur ett utsågat trästycke deformeras när fukt- och temperaturförhållandena ändras. Orsaken är trämaterialens anisotropa egenskaper. Krympning och expansion sker olika mycket i olika riktningar. Bildkälla: [41].

Tabell 2.1: Medelvärden på hållfastheten i MPa för små felfria provkroppar av gran (*picea abies*) enligt Berbom Dahl [3]. Karakteristiska hållfasthetsvärden enligt EN 338:2009 [16]. f_{Lt} och f_{Lc} betecknar drag- respektive tryckhållfastheten parallellt fibrerna. f_{Rt} , f_{Rc} , f_{Tt} och f_{Tc} definieras på analogt vis. f_{LR} , f_{LT} och f_{RT} betecknar skjuvhållfastheter.

Källa	f_{Lt}	f_{Lc}	f_{Rt}	f_{Rc}	f_{Tt}	f_{Tc}	f_{LR}	f_{LT}	f_{RT}
[3]	63	29	4,9	3,6	2,8	3,8	6,1	4,4	1,6
[16] (C16)	10	17	0,4	2,2	0,4	2,2	3,2	3,2	
[16] (C24)	14	21	0,4	2,5	0,4	2,5	4,0	4,0	
[16] (C40)	24	26	0,4	2,9	0,4	2,9	4,0	4,0	

beroende på var i stammen det sågas ut.

På grund av ovan nämnda defekter fås en relativt stor spridning i hållfastheten. Genom att sortera virket i hållfasthetsklasser kan spridningen kontrolleras och materialet utnyttjas till fullo. Eftersom kvistar har störst inverkan på hållfastheten är det vanligt med visuell sortering med antingen manuell eller maskinell metod [25]. Det har också visat sig i en mängd experiment att storleken på trästycket har betydelse för hållfastheten som tenderar att minska med tilltagande provkroppsvolym [48]. Förklaringen brukar bygga på sannolighetsteoretiska resonemang med Weibullteori och går ut på att antalet defekter ökar med tilltagande volym [25, 11]. Se emellertid [52] för en alternativ förklaringsmodell. De främsta svagheter hos trä som byggnadsmaterial är enligt [3]

- de mekaniska egenskapernas relativt stora variation
- låg styvhet och hållfasthet vinkelrätt fiberriktningen
- brandsäkerheten
- ljudisoleringen
- risken för röta och biologiska angrepp i framför allt utomhusmiljö

2.2.3 Europeiska normer

Fukt har stor inverkan på verknings sättet och därför indelas bärverk av trä enligt europeiska normer (Eurocode 5) i tre kategorier beroende på

Tabell 2.2: Medelvärden på styvhetsparametrarna i MPa för små felfria provkroppar av gran (*pinus abies*) enligt Berbom Dahl [3]. Karakteristiska värden på styvhetsparametrarna enligt EN 338:2009 [16]. E , G och ν betecknar elasticitetsmodul, skjvmodul respektive tvärkontraktionstal.

Källa	E_L	E_R	E_T	G_{LR}	G_{LT}	G_{RT}	ν_{LR}	ν_{RL}	ν_{LT}	ν_{TL}	ν_{RT}	ν_{TR}
[3]	9 040	790	340	640	580	30	0,5	0,11	0,66	0,06	0,84	0,34
[16] (C16)	8 000	270	270	500	500							
[16] (C24)	11 000	370	370	690	690							
[16] (C40)	14 000	470	470	880	880							

vilken sorts miljö och klimat det förväntas befina sig i under brukstiden [15]. Den första så kallade klimatklassen innebär att den relativa fuktigheten överstiger 65 % högst under några veckor per år. I klimatklass II är gränsen istället 85 % och klimatklass III definieras för sådan miljö som varken tillhör klimatklass I eller II.

Densiteten ρ_m varierar i olika delar i stammen och mellan olika skogsbestånd. Berbom Dahl [3] provbelastade ett stort antal små, felfria trästycken av gran som befann sig i jämvikt med 12 % fuktkvot och fann att de i medeltal hade densiteten $\rho_m = 398 \text{ kg/m}^3$ med medelhållfastheter enligt första raden i Tabell 2.1 och styvhetsvärden enligt första raden i Tabell 2.2. Dessa hållfasthetsvärden skiljer sig markant från dem i de europeiska normerna som för jämförelsens skull finns med i tabellerna. Det beror på att normerna anger det så kallade karakteristiska värdet som motsvarar 5-percentilen i en statistisk fördelning. Värdet är valt så att den statistiska sannolikheten för att erhålla ett konstruktionsvirke med lägre hållfasthet än den utfästa ska vara minimal. Vanligen bestäms den karakteristiska böjhållfastheten och densiteten med icke- destruktiva provningar. Övriga hållfasthetsparametrar härleds sedan ur dessa med hjälp av vissa mer eller mindre empiriskt belagda samband [16].

Konstruktionsvirke sorteras i så kallade C-klasser. Till exempel betyder C16 ett konstruktionsvirke som i genomsnitt i 98 fall av 100 har en hållfasthet större än 16 MPa i böjning. De moderna byggnormerna är föränderliga och uppdateras ständigt allt eftersom nya forskningsresultat framkommer. Detta gäller inte minst de karakteristiska hållfasthetsvärdena för trä. Till exempel sänktes nyligen den karakteristiska tryckhållfastheten vinkelrätt fibrerna från cirka 8 till mellan 2-3 MPa när Eurocode 5 implementerades istället för den dåvarande svenska normen Boverkets byggregler (BKR). Dessutom skiljer sig normer och karakteristiska värden mellan europeisk, nordamerikansk och australiensisk/nya zeeländsk norm. Att bygga i trä är ingen exakt vetenskap.

Framgång bygger snarare på en djup förståelse för träets komplexa verkningssätt och hur det samverkar med övriga delar i ett byggnadssystem.

2.2.4 Materialmodellering

Betrakta ett litet, homogent och felfritt trästycke och antag att det är ortotrop med huvudriktningar enligt Figur 2.6. Då kan Hookes lag i tre dimensioner uppställas.

$$\begin{aligned}\boldsymbol{\sigma} &= \mathbf{D}\boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &= [\epsilon_{LL} \ \epsilon_{RR} \ \epsilon_{TT} \ \gamma_{LR} \ \gamma_{LT} \ \gamma_{RT}]^T \\ \boldsymbol{\sigma} &= [\sigma_{LL} \ \sigma_{RR} \ \sigma_{TT} \ \tau_{LR} \ \tau_{LT} \ \tau_{RT}]^T\end{aligned}$$

Styvhetsmatrisen \mathbf{D} definieras som

$$\mathbf{D} = \mathbf{C}^{-1}, \quad \mathbf{C} = \begin{bmatrix} \frac{1}{E_L} & -\frac{\nu_{RL}}{E_R} & -\frac{\nu_{TL}}{E_T} & 0 & 0 & 0 \\ -\frac{\nu_{LR}}{E_L} & \frac{1}{E_R} & -\frac{\nu_{TR}}{E_T} & 0 & 0 & 0 \\ -\frac{\nu_{LT}}{E_L} & -\frac{\nu_{RT}}{E_R} & \frac{1}{E_T} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{LR}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{LT}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{RT}} \end{bmatrix} \quad (2.1)$$

med elasticitets- och skjuvmodulerna E_L, E_R, E_T och G_L, G_R, G_T och tvärkontraktionstalen ν_{ij} . Styvhetsmatrisen blir symmetrisk om följande samband antas [11]

$$\frac{\nu_{RL}}{E_R} = \frac{\nu_{LR}}{E_L}, \quad \frac{\nu_{TL}}{E_T} = \frac{\nu_{LT}}{E_L}, \quad \frac{\nu_{TR}}{E_T} = \frac{\nu_{RT}}{E_R}$$

I ett plan som är parallellt med L-axeln (jämför Figur 2.6) kan ett plant spännings- eller töjningstillstånd förenklat ansättas. Då brukar trä modelleras som transversellt isotropt vilket betyder att man inte längre skiljer på egenskaperna i olika riktningar transversellt L-axeln. Plant spänningstillstånd betyder att spänningarna vinkelrätt planet försummas (men inte tvärkontraktionen) och äger större giltighet ju slankare sektion som betraktas. Plant töjningstillstånd definieras analogt [38].

2.3 Brottmekanik

Den som slitit itu ett tygstycke eller klyvt ved med en yxa har erfarit hur en spricka sänker hållfastheten i ett material. Ett urtag medför på liknande sätt en försvagning av materialet. För att förklara detta används brottmekanik.

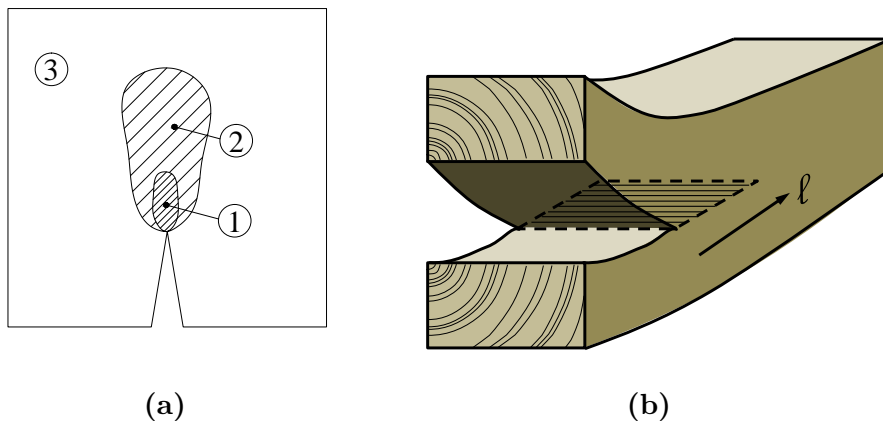
2.3.1 Inledande definitioner

Betrakta en fast kropp med en befintlig spricka, se Figur 2.8a. Med en sprickbildning menas att delar av kroppen förlorar kontakt med varandra. Närmast sprickspetsen återfinns en *brottzon* (1) vars storlek ofta antas vara liten i förhållande till kroppens övriga dimensioner. I duktila material fås plasticering (2) i närheten av sprickan. Långt från sprickspetsen utanför brottzon och flytzone arbetar materialet elastiskt (3). Ett duktilt material är ett material som inte är sprött. Ett sprött material går till brott medan det fortfarande arbetar elastiskt. En sprickspets ska inte nödvändigtvis tolkas som en punkt såsom föranleds av Figur 2.8a. Snarare avses en mängd punkter längs en linje vilken utgör sprickans front. *Brottytan* är de nya materialytor som uppstår när sprickan tillväxer och fås genom att svepa en linje framför sprickspetsen, vilket illustreras i Figur 2.8b.

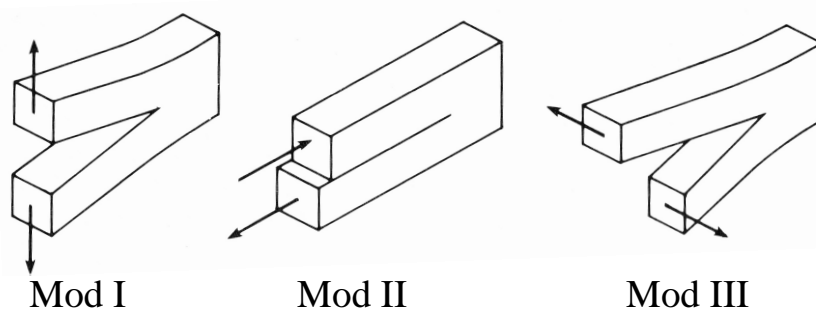
Spricktillväxt delas geometriskt sett upp i tre olika så kallade *brottmoder*. Mod I innebär att brottytorna förskjuts symmetriskt och parallellt med sina normaler. Mod II och III innebär antisymmetrisk förskjutning av brottytorna parallellt med en tangentvektor i brottytans plan, jämför Figur 2.9. Tangentvektorn är i ena fallet vinkelrät mot sprickspetsfronten och i andra fallet parallell med den. En spricktillväxt kan vara stabil eller instabil. Instabil spricktillväxt medför oftast mekanisk kollaps [23].

2.3.2 Spänningstillståndet vid sprickspetsen med antagande om linjärelasticitet

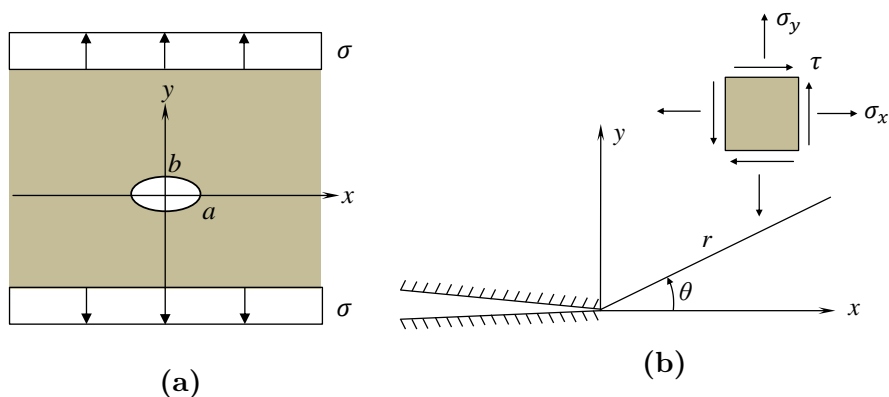
Betrakta en skiva med oändlig utsträckning som belastas enaxiellt i oändligheten. Antag att materialet arbetar linjärelastiskt och inför ett elliptiskt hål med halvaxlarna a och b och koordinatsystem såsom i Figur 2.10a. Spänningar och töjningar i närheten av det elliptiska hålet kan



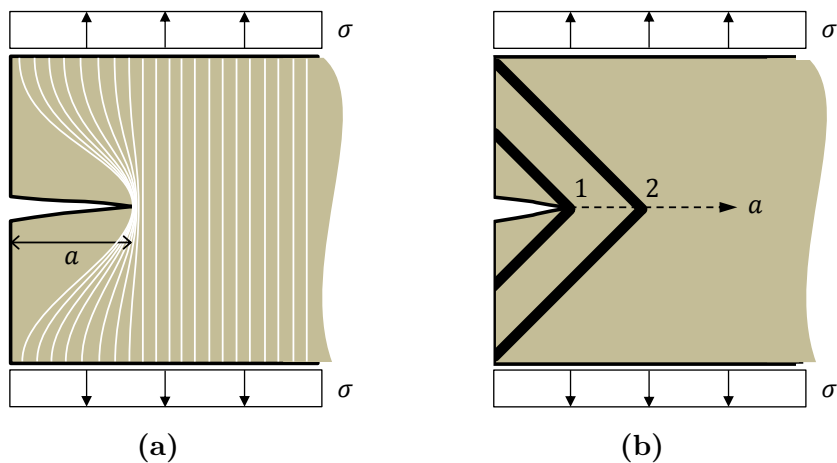
Figur 2.8: a) En fast kropp med spricka. Närmast sprickspetsen återfinns brottzonen (1). I duktila material fås en flytzone (2). Långt från sprickspetsen arbetar materialet elastiskt (3). Figur efter [34]. b) Brottytan erhålls genom att svepa en linje framför den tillväxande sprickspetsfronten.



Figur 2.9: Geometriska moder vid spricktillväxt. Mod I innebär symmetrisk förskjutning av brottytorna parallellt med normalen. Mod II och III innebär antisymmetrisk förskjutning av brottytorna parallellt med en tangentvektor i brottytans plan. Tangentvektorn är i ena fallet vinkelrät mot sprickspetsfronten och i andra fallet parallell med den. Bildkälla: [23].



Figur 2.10: a) Oändligt utsträckt skiva med elliptiskt hål och halvaxlarna a och b . Skivan belastas enaxiellt i oändligheten. Figur efter [34]. b) Spänningstillståndet nära spetsen av en skarp spricka ges av Westergaards lösning. r är det radiella avståndet från sprickspetsen. θ är vinkeln mellan sprickans symmetrilinje och strålen från sprickspetsen genom den betraktade punkten. Figur efter [23].



Figur 2.11: a) Betrakta en skiva med en skarp spricka med längden a . I figuren antyds spänningsfältet som uppstår i materialet på grund av den yttre lasten σ . I närheten av en tillväxande spricka förändras spänningsfältet så att delar av materialet avlastas. Då frigörs elastisk töjningsenergi. b) Den materialvolym som avlastas när sprickan tillväxer kan uppskattningsvis avgränsas med räta linjer. Då inses att en större mängd töjningsenergi frigörs när sprickan tillväxer från läge (2) än från läge (1). Slutsatsen är att små sprickor är mindre farliga än stora med avseende på instabil tillväxt. Figurer efter [44].

studeras med hjälp av analytiska funktioner¹ vilket ger att spänningen σ_y i punkten $(x, y) = (a, 0)$ är

$$\sigma_y = \sigma \left(1 + 2\frac{a}{b}\right)$$

det vill säga att spänningarna växer obegränsat då halvaxeln $b \rightarrow 0$ och brottzonen är punktformig [34].

Ersätt istället det elliptiska hålet med en rak spricka med längd a och med skarp sprickspets och punktformig brottzon så som i Figur 2.10b. Då ges spänningstillståndet i närheten av sprickspetsen av Westergaards lösning [34]. Om avståndet från sprickspetsen är r och vinkeln till sprickans symmetrilinje θ så blir

$$\begin{cases} \sigma_y &= \sigma \sqrt{\frac{a}{2r}} \cos \frac{\theta}{2} \left(1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2}\right) + \dots \\ \sigma_x &= \sigma \sqrt{\frac{a}{2r}} \cos \frac{\theta}{2} \left(1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2}\right) + \dots \\ \tau &= \sigma \sqrt{\frac{a}{2r}} \sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2} + \dots \end{cases}$$

där högre ordningens termer är försumbara för $r \ll a$. Spänningarna växer obegränsat då $r \rightarrow 0$. Spänningstillståndet i ett ortotrop material är något mer invecklat att beskriva, den intresserade läsaren hänvisas till exempelvis [21]. Slutsatsen är hur som helst densamma eftersom $\frac{1}{\sqrt{r}}$ -beroendet är gemensamt för spänningarna i isotropa och ortotropa material.

Det är orealistiskt att påstå att spänningarna är obegränsat stora vid sprickspetsen och samtidigt anta att hållfastheten är ändlig. Motsägelsen beror på att det i verkligheten inte finns några perfekt linjärelastiska material med matematiskt skarpa sprickspetsar. Alla fasta kroppar är i verkligheten behäftade med mikrosprickor. Brottvillkoret för en konstruktionsdel bygger vanligen på spänningstillståndet i det mest belastade snittet. Men eftersom spänningarna intill en sprickspets i teorin är obegränsade måste brottvillkoret för detta fall baseras på något annat än spänning.

2.3.3 Principen om energijämvikt

På 1920-talet studerade den brittiske ingenjören A. A. Griffith spröda brott i keramiska material. Han insåg att när en befintlig spricka tillväxer,

¹Funktionen $f(z)$ säges vara analytisk i en punkt $z \in \mathbb{C}$ där \mathbb{C} betecknar det komplexa planet om f är deriverbar i varje punkt i en omgivning till z [34].

kommer en del av materialet runt sprickan att avlastas med följden att elastisk töjningsenergi frigörs [44]. Idén illustreras i Figur 2.11a. Griffith antog att den frigjorda energin omsätts i bildandet av nya brottytor [44]. För en idealt spröd provkropp som belastas under nära nog statiska förhållanden kan antagandet motiveras på så vis: låt S beteckna energin per ytenhet av en brottyta och använd energilagen för att teckna den totala energin [20]

$$E_{tot} = U_e + E_k - W + S \quad (2.2)$$

där

E_{tot}	Total energi
U_e	Elastisk töjningsenergi
E_k	Kinetisk energi
W	Arbete
S	Energi per ytenhet av en brottyta

Differentiering av uttrycken i (2.2) med avseende på brottytan A ger

$$0 = \frac{\partial U_e}{\partial A} + \frac{\partial E_k}{\partial A} - \frac{\partial W}{\partial A} + 2\gamma$$

där γ med Griffiths terminologi är den energi per ytenhet som fordras för att bilda en fri brottyta. Faktorn 2 uppträder därför att en spricka ger upphov åt två fria brottytor. Om spricktillväxten sker under nära nog statiska förhållanden, det vill säga utan alstring av kinetisk energi, så är $\frac{\partial E_k}{\partial A} = 0$. Villkoret för uppsprickning blir

$$\frac{\partial W}{\partial A} - \frac{\partial U_e}{\partial A} = 2\gamma$$

Griffith föreställde sig att den frigjorda energin förbrukas på att bryta de atom- och molekylobindningar som sammanhåller materialet före sprickans tillkomst. Detta är sant för ett idealt sprött material. Vi har alltså Griffiths postulat

Griffiths postulat När en befintlig spricka tillväxer omsätts frigjord töjningsenergi i bildandet av fria brottytor

Senare föreslog Irwin och Orowan att den energi som krävs för att bryta atombindningarna i duktila material endast utgör bråkdelen av den frigjorda energin [44]. De menar att frigjord energi förbrukas genom plasticering och materialomvandling i brottzonen. Den energimängd som

kan omsättas i närheten av sprickspetsen kallar Irwin och Orowan för G_c : kritisk frigörelse av töjningsenergi². Med andra ord gäller att

“Frigjord töjningsenergi” = 2γ + “flytning och materialomvandlingseffekter”

G_c är en materialspecifik parameter som beskriver motståndet mot instabil spricktillväxt och kan jämföras med den så kallade *brottsegheten* K_c . Med Westergaards lösning visas att för ett isotropt material är $G_c = \frac{K_c^2}{E}$ [34]. Så länge den frigjorda töjningsenergin understiger den kritiska energifrigörelsen G_c , bibehåller sprickan sin befintliga längd. Först när den kritiska gränsen överskrids, och det inte finns möjlighet för materialet att omsätta mera energi genom plasticering, tillväxer sprickan. Villkoret för uppsprickning blir alltså

$$\frac{\partial}{\partial A} (W - U_e) = G_c \quad (2.3)$$

Exempel 2.3.1 Betrakta en fast inspänd och dragbelastad kropp med urtag och tvärsnittsbredden b . Vid urtagets hörn finns en spricka med längden a . Inför ett koordinatsystem såsom i illustrationen nedan. Arbetet som den konstanta lasten P uträttar då kraftens angreppspunkt förskjuts δ är

$$W = \int_0^\delta P \, du = P\delta$$

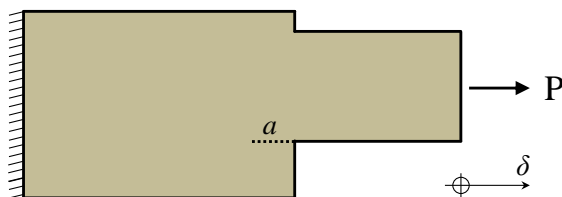
Den elastiska töjningsenergi som lagras när kroppen deformeras på grund av lasten är

$$U_e = \frac{P\delta}{2}$$

Insättning av ovanstående två uttryck i brottvillkoret (2.3) ger

$$\frac{d}{da} \left(\frac{1}{b} \cdot \frac{P\delta}{2} \right) = G_c$$

²På engelska: *critical strain energy release rate*

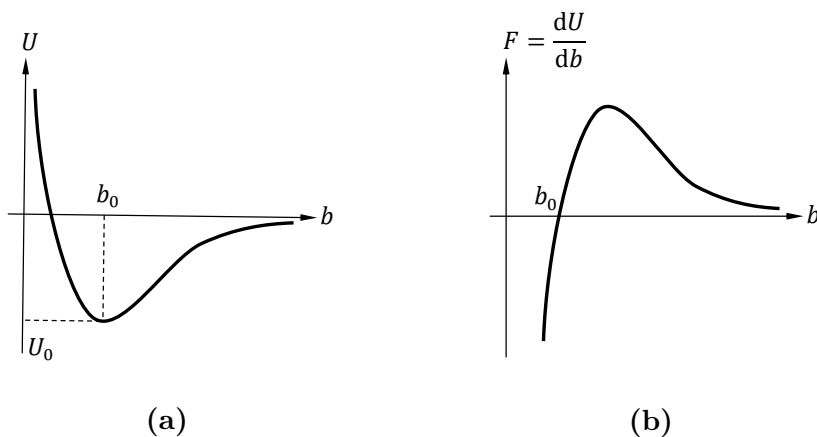


Genom att uttrycka δ i a skulle bärförmågan med avseende på uppsprickning P kunna bestämmas.

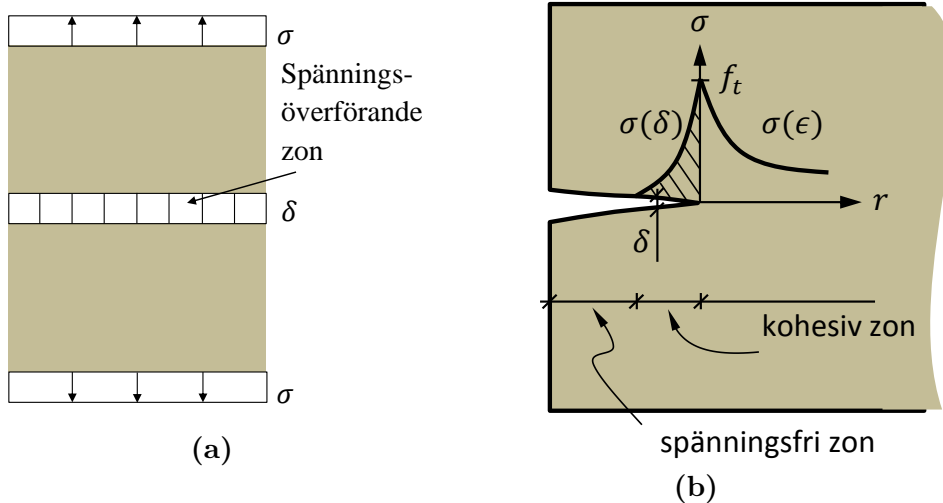
Griffiths hypotes förklarar varför små föremål tenderar att vara brottsegare än stora. Betrakta kroppen i Figur 2.11b. Den materialvolym som avlastas vid spricktillväxt kan förenklat avgränsas med räta linjer. Uppenbarligen frigörs mer energi när sprickan tillväxer från läget (2) än när den tillväxer från läget (1). Därför är det sprickans absoluta längd som avgör hur "farlig" den är: en tillräckligt lång spricka kan leda till instabilitetsbrott.

2.3.4 Kohesiv zon

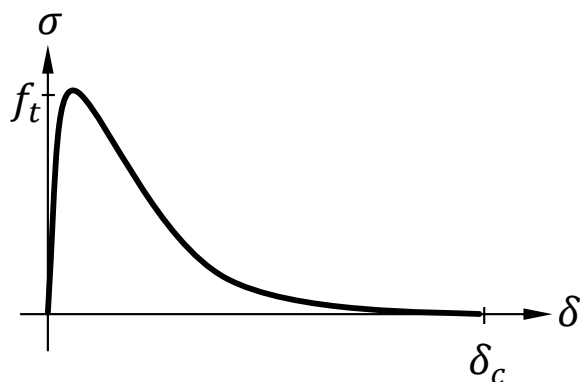
Enligt Barenblatt hålls brottzonen samman av kohesion [39]. För ett idealt sprött material kan idén troliggöras genom att studera den atomära potentialkurvan, vars principiella utseende illustreras i Figur 2.12a. Med figurens beteckningar gäller att kraften $F = -\frac{dU}{db}$ så att kraftförskjutningskurvan mellan två atomer i princip ges av Figur 2.12b. Uppenbarligen finns en kritisk separation b_c , utöver vilken atomparets kohesion är försumbart liten. Ytan under grafen i Figur 2.12b kan tolkas som brottenergi. Med brottenergi menas den energi som krävs för att åstadkomma spricktillväxt i ett material. Om materialet utanför brottzonen är elastiskt och brottzonen fullt utvecklad (det vill säga vid spricktillväxt förflyttas brottzonen utan att utvecklas) överensstämmer den kritiska energifrigörelsen G_c med brottenergin G_f . Petersson [39] bygger på Barenblatts antagande och ersätter brottzonen med en kohesiv zon som består av en fiktiv spänningsöverförande spricka. Förmågan att överföra spänning beror på zonens bredd. På så vis är det möjligt att studera brottzonens propagering och sprickans tillväxt. Idén antyds genom att studera verknings sättet hos ett direkt dragprov av en stav. Staven



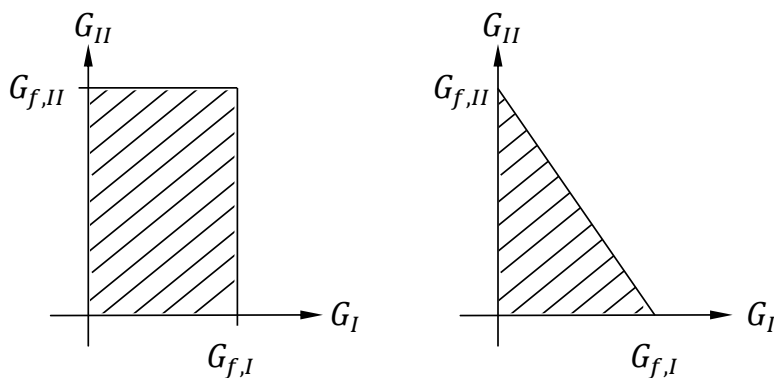
Figur 2.12: a) Potentialkurva för ett atompar. Vid avståndet b_0 återfinns jämvikt. b) För att separera ett atompar från sitt jämviktstillstånd b_0 fordras en kraft F som antas försumbart liten för tillräckligt stora avstånd b .



Figur 2.13: a) Spänningsöverförande zon med bredd δ i en dragbelastad stång. b) Principiell spänningsfördelning nära sprickspetsen med kohesiva zonmodellen. I brottzonen beräknas spänningar utifrån förskjutningar medan ett vanligt spännings-töjningssamband används utanför brottzonen.



Figur 2.14: Samband mellan spänning och förskjutning i den kohesiva zonen.



Figur 2.15: Två enkla samband mellan de kritiska töjningsenergierna i Mod I och Mod II. Ytan under grafen till vänster ges av mängden

$$\{(G_I, G_{II}) : 0 \leq G_I \leq G_{f,I} \text{ och } 0 \leq G_{II} \leq G_{f,II}\}$$

och innebär att brottenergierna ej samspelar. Ytan under grafen till höger ges av

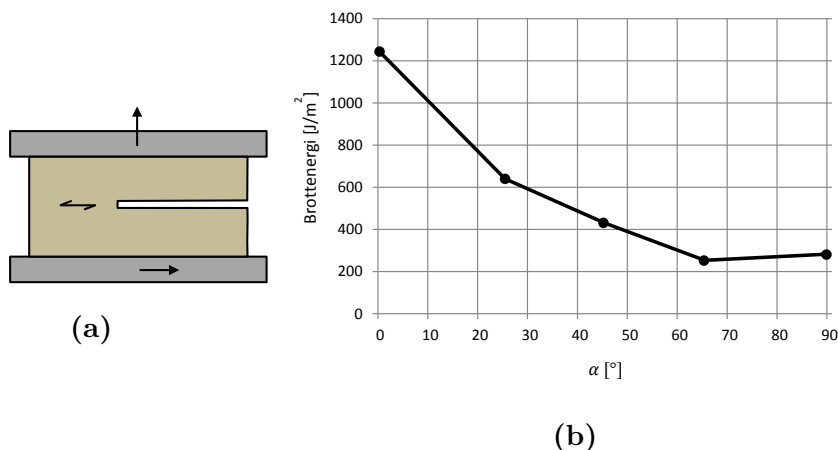
$$\{(G_I, G_{II}) : 0 \leq G_{f,I}G_{II} + G_{f,II}G_I \leq G_{f,I}G_{f,II} \text{ och } 0 \leq G_I\}$$

och innebär ett linjärt samband mellan brottenergierna.

Tabell 2.3: Värden på brottenergi och hållfasthet för gran med fuktkvot 14 % enligt sammanställning i [11].

	G_f [J/m ²]	f_t [MPa]
Mod I	283	2.9
Mod II	1240	9.8

antas vid brott gå itu längs en transversell linje som indelar staven i två delar. Den kohesiva zonen omfattas av det transversella snittet. I Figur 2.13a illustreras detta genom att kohesionszonens bredd δ växer med stigande belastning. Man kan indela stavens totala förlängning i en del som beror på elastisk töjning *utanför* kohesionszonen och en annan del som beror på olinjär förskjutning *innanför* kohesionszonen. Spänningarna varierar därmed i princip enligt Figur 2.13b. Ett möjligt samband mellan spänning och förskjutning i brottzonen illustreras i Figur 2.14. För små förskjutningar är sambandet linjärt enligt figuren och materialet arbetar elastiskt. När hållfastheten uppnås börjar brottzonen propagera. Den oelastiska delen av kurvan kännetecknas av uppkomsten av mikrosprickor och flytning i brottzonen. Emellertid kan materialet fortfarande överföra spänning. Till slut, vid en kritisk öppningsbredd δ_c , uppspricker materialet. Den kritiska öppningen δ_c kan beräknas med utnyttjande av kända värden på brottenergin G_f , hållfastheten f_t och utseendet på $\sigma - \delta$ -kurvan. Eftersom brottzonen framför ett urtag vanligen propagerar på grund av dragspänning, kan den antas ha egenskaper liknande den kohesiva zonen vid ett direkt dragprov. Peterssons modell bygger på ren Mod I förskjutning. Förskjutningarna vid ett urtag präglas emellertid av både Mod I och Mod II. För att implementera kohesiva zonmodellen på urtag, antas att kohesionszonen även förmår överföra skjuvspänning motsvarande den som uppstår vid ren Mod II. Eftersom både brottenergi och hållfasthet varierar med riktningen i trä så blir den kritiska bredden i kohesionszonen på grund av Mod II en annan än på grund av Mod I. Det bör beaktas att brottenergierna (eventuellt) samspelar. Två enkla samspel illustreras i Figur 2.15. Sammanfattningsvis består brottzonen av en fiktiv spänningsöverförande spricka eller kohesiv zon vilken överför spänning transversellt och parallellt den fiktiva sprickytan. Brottenergin samspelar eventuellt i Mod I och Mod II enligt något givet samband. Så snart en del av den fiktiva sprickan öppnar sig eller förskjuts mer än vad som medges av interaktionssambandet, mister den all kohesiv förmåga och övergår i en verklig spricka. Vad som fordras för att implementera



Figur 2.16: a) Illustrerar en testuppställning som kan användas för att uppmäta brottenergin i Mod I, Mod II eller någon kombination. Bildkälla: [11] b) Uppmätt brottenergi som funktion av modkombinationen hos gran med fuktkvot 14 % enligt [46]. $\alpha = 0$ motsvarar ren Mod II och $\alpha = 90$ motsvarar ren Mod I.

den kohesiva zonmodellen är uppgift om hållfasthet, brottenergi och den olinjära last-förskjutningskurvans form samt samspelseffekten mellan brottenergierna i Mod I och Mod II.

2.3.5 Så uppmäts brottenergi

I Figur 2.16a visas en testuppställning som kan användas för att bestämma brottenergin (fler exempel finns i [11]). Testuppställningen används för att mäta last-förskjutningskurvan som sedan jämförs med uträttat arbete av last och egenvikt. Danielsson [11] har sammanställt uppmätta brottenergier från tio olika källor. För gran (*pinus abies*) med fuktkvot 14 % ges med hänvisning till [46] värden på brottenergier och hållfastheter i Mod I och Mod II enligt Tabell 2.3. Samma källa ger brottenergin som funktion av modkombinationen vilket visas i Figur 2.16b.

2.3.6 Ökända sprickor i modern tid

Comet blev år 1952 världens första passagerarjetflygplan i kommersiell trafik men belades redan år 1954 med flygförbud efter flera totalhaverier. Tillverkare var brittiska De Havilland. I Figur 2.17a återges ett fotografi av flygplanet, som på sin tid betraktades som en teknisk landvinning.

Haveriutredningen som följde var omfattande och inbegrep att bärga många vrakdelar från havsbotten och att trycksätta ett helt flygplan i en stor vattenbassäng. Winston Churchill uppges ha sagt: "The cost of solving the Comet mystery must be reckoned neither in money nor in manpower." [4] Roylance [44] beskriver utredningen som en av de mest omfattande i ingenjörshistorien. När flygplanet befann sig i luften orsakade övertrycket i kabinen att små sprickor bildades vid nitarna på flygplanskroppen. På grund av mekanisk utmattning tillväxte sprickorna. Den kritiska spricklängden överstegs med instabil spricktillväxt som följd när en från början liten spricka fortplantat sig till kanten av ett fönster eller en lucka. När *Comet* tjänstgjort i två år hade den på grund av utmattning uppnått sin livslängd. För att lösa detta monterades tvärgående band längs flygplanskroppen. Utmattningssprickor var en av orsakerna till att taket till första klass på en Boeing 737 slets av under en flygning år 1988, se Figur 2.17b [36, 44]. *Schenectady* var ett 152 meter långt tankfartyg av modell T2, byggt i Portland i delstaten Oregon i USA, som den 16:e Januari 1943 återvände från en lyckad provseglat. Vattnets temperatur uppges ha varit 4 °C och luften vindstilla, när skeppet senare på natten utan förvarning rämnade mitt itu. Det skedde med en smäll som lär ha hörts mer än en och en halv kilometer bort. Ett fotografi av den rämnade farkosten finns visas i Figur 2.18a. Haveriet berodde på sprickbildning i stålet [49, 57].

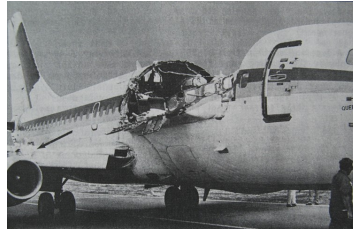
Detta är endast ett litet urval av exempel på katastrofala sprickor. Antalet haverier och olyckor i modern tid är dessvärre enormt. För att ta ytterligare ett exempel: under andra världskriget byggdes i USA cirka 2700 så kallade *Libertyskepp*, lätt bestyckade transportfartyg. Mellan 5-30 % av dessa drabbades av katastrofala brott i skrovet på grund av sprickor. I Figur 2.18b syns ett *Libertyskepp* som fått aktern avsliten [57]. I "The economic effects of fracture in the United States of America" från 1983 uppskattas de sammanlagda kostnaderna i USA för brott i konstruktioner på grund av sprickor till 119 miljarder dollar *per år* i 1982 års dollarkurs [40].

2.3.7 Sprickor i populärkulturen

Amerikanen Nevil Shute (1899-1960) har författat ett tjugotal skönlitterära verk. De flesta finns i svensk översättning. I romanen *Utan återvändo* från 1948, möter vi den självupptagne Theodore Honey som forskar i mekanik på Royal Aircraft Establishment i Farnborough. Theodore är

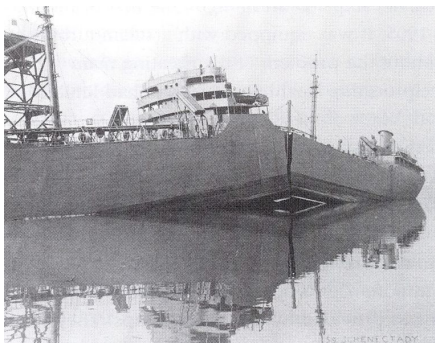


(a)



(b)

Figur 2.17: a) De Havillands *Comet-1* betraktades som en teknisk landvinning på sin tid men belades redan år 1954 med flygförbud efter att sprickor i flygplanskroppen vållat flera totalhaverier. På bilden Aeromaritime de Havilland DH-106 Comet. Notera de rektangulära fönstren. Bildkälla: [55] b) Boeing 737 efter Aloha Airlines Flight 243 år 1988 då utmattningsprickor orsakade att taket till första klass slets av. Bildkälla: [35]



(a)



(b)

Figur 2.18: a) Tankfartyget *Schenectady* som byggdes i USA år 1943 hade just återvänt från en provseglat utan förebud när den senare på natten rämnade mitt itu. Bildkälla: [56] b) Under andra världskriget byggdes cirka 2700 *Libertyskepp* i USA för att frakta gods över Atlanten. Här syns *SS John P Gaines* som fått aktern avsliten på grund av sprickor. Bildkälla: [58]

änkling och bor tillsammans med sin dotter som han känslomässigt distanserar sig från. Genom noggrann beräkning drar han slutsatsen att passagerarflygplanet *Raindeer* som tillverkas på RAE endast kommer att hålla för 1400 flygtimmar, varefter svansfenan förmodas brista på grund av mekanisk utmattning och instabil spricktillväxt. Eftersom en flotta av *Raindeer* redan börjat tjänstgöra i trafik befinner sig många människoliv i fara. Men detta går inte upp för antihjälten Theodore förrän han själv befinner sig ombord på en *Raindeer* på tjänsteresa till Kanada och uppnår katarsis. På den långa transatlantiska färden utspelar sig ett existentiellt drama där Theodore tvingas lyfta blicken bortom de mekaniska sambanden och ta itu med villkoren för sin egen tillvaro. Theodore löser med framgång krisen genom att axla en fadersroll han hittills undvikit och flygplanet landar oskadat i den nya världen. Med facit i hand kan det konstateras att fiktionens värsta farhåga besannades ett fåtal år senare i och med *Comets* totalhaverier. Romanen har på grund av behandlingen av ett solidmekaniskt ämne och den egendomliga förmågan att förutspå något som inträffade i verkligheten erövat en informell kultstatus bland mekanikingenjörer enligt Roylance [44]. Romanen filmatiserades 1951 med James Stewart i huvudrollen och Marlene Dietrich i birollen. *BBC* [2] har också gjort en radioserie av romanen.

2.4 ABAQUS

2.4.1 Allmänt

ABAQUS är en mjukvara som lanserades år 1978 av Hibbit, Karlsson & Sorensen Inc [42]. Sedan oktober 2005 ägs det av Dassault Systèmes Simulia Corp och kallas ABAQUS FEA [31]. Abaqus är en referens till den klassiska kulramen och FEA står för Finite element analysis. ABAQUS FEA (hädanefter förkortat ABAQUS) är en programvarusvit som inbegriper följande fyra delar [12, 13]

- ABAQUS/CAE kan lösa såväl statiska som dynamiska finita elementproblem med både linjär och olinjär materialmodellering. Det är det mest kompletta gränssnittet i sviten och används i det här examensarbetet. CAE står för “complete ABAQUS environment”. Akronymet brukar i andra sammanhang stå för “computer aided engineering”.
- ABAQUS Standard möjliggör implicita lösningsmetoder för både statiska och dynamiska finita elementproblem med såväl linjär som olinjär materialmodell. Det finns också tilläggs paket för analys av till exempel undervattenskonstruktioner.
- ABAQUS Explicit möjliggör explicita lösningsmetoder för olinjära transienta finita elementproblem.
- ABAQUS/CFD är utvecklat för att lösa flödesdynamiska problem med finita elementmetoden. CFD står för “computational fluid dynamics”.

Användandet av ABAQUS/CAE är indelat i tre huvudsteg.

1. I *preprocessen* definierar användaren modellens geometri, materialegenskaper, elementnät, randvillkor med mera. Detta kan göras antingen 1) i det grafiska gränssnittet (GUI: Graphical User Interface) eller 2) genom att skriva en input-fil (`.inp`) som är en vanlig ASCII-kodad textfil eller 3) genom implementering av Python-skript eller 4) implementering av C-skript. Historiskt sett är metod (2) standard. Det finns en omfattande verktygslåda av funktioner som användaren kan anropa i input-filen för att skapa komplexa FE-modeller. I ABAQUS-manualen som finns tillgänglig på Internet finns förklaringar, teori och exempel på i stort sett samtliga funktioner [13].

2. I andra steget analyseras modellen av kärnan i ABAQUS och lösningar till jämviktsekvationerna sökes. Resultatet skrivs till en outputdatabas-fil (.odb).
3. I *postprocessen* kan resultaten i outputdatabasen analyseras och bearbetas. Med det grafiska gränssnittet finns möjlighet att visualisera lösningarna och erhålla information om till exempel krafter och förskjutningar för enskilda noder och element.

Implementering av Python-skript är ett kraftfullt verktyg för att skräddarsy modelleringen och bearbetningen av analysresultaten. Implementering av C-skript medger en i vissa fall effektivare numerisk beräkning (mindre datortid) men skiljer sig annars inte i princip från metoden med Python-skript.

2.4.2 Att implementera Python-skript i ABAQUS

Vad är Python?

Python är ett så kallat "general-purpose" objektorienterat programmeringsspråk som lanserades 1989 av Holländaren Guido van Rossum [43]. Namnet inspirerades van Rossum till då han läste manuskriptet till Monty Python's Flying Circus, en BBC-serie från 1970-talet [19]. Med objekt avses inom datavetenskapen en klass som innehåller datafält och metoder [27]. Enligt TIOBE Programming Community Index positionerar sig Python bland de tio populäraste programmeringsspråken år 2013 [50]. Python har använts av bland andra Google, Yahoo!, CERN och NASA och ingår som en standardkomponent i flera operativsystem, bland andra Linux och OS X [18, 9, 45]. Python positionerar sig som ett läsbart och användarvänligt språk, vilket kommer till uttryck i 1) den så kallade off-side-regeln³ som innebär att deklarerade block skall indenteras. 2) Kodorden liknar dem i C och Java, men oftare används i Python engelska ord, exempelvis `and`, `or` och `not` i stället för `&&`, `||` och `!`. 3) Metoder anropas explicit med `instance.method(argument)` i stället för `Class.method(instance, argument)` såsom i C++ och Java; en syntaktisk omskrivning vars syfte är att öka läsbarheten. Alex

³Off-side-regeln definierades så här av Peter J. Landin år 1966: "Any non-whitespace token to the left of the first such token on the previous line is taken to be the start of a new declaration." [28]

Martelli [32] sammanfattar ambitionen i Python med orden: "To describe something as clever is not considered a compliment in the Python culture."

Vägledning för nybörjare

Använd GUI:n och låt dig inspireras av replayfilen `abaqus.rpy` som återfinns i ABAQUS temporärkatalog (till exempel `C:\SIMILIA\Temp`). Så snart användaren utför en handling i GUI:n skrivs motsvarande Python-anrop till replay-filen.

Exempel 2.4.1 Med GUI:n skapar användaren en ny modell och en rektangulär 2D-kropp med dimensionerna 4x1.2. Replay-filen visar

```
Mdb()
s = mdb.models['Model-1'].ConstrainedSketch(
    name='__profile__', sheetSize=5.0)
s.rectangle(point1=(0.0, 0.0), point2=(4, 1.2))
p = mdb.models['Model-1'].Part(
    name='Part-1', dimensionality=TWO_D_PLANAR,
    type=DEFORMABLE_BODY)
p.BaseShell(sketch=s)
```

Förklaring:

<code>Mdb()</code>	Med första raden skapas en ny modell. Notera att metoder som skapar nya delar (till exempel en ny modell, ett nytt material, ett nytt interaktionssamband med mera) har namn som börjar med stor bokstav.
<code>ConstrainedSketch()</code>	Andra raden skapar en ny rityta med storlek 5.0 och som i det här fallet benämns <code>'__profile__'</code> . Ritytan tilldelas variabeln <code>s</code> . Kom ihåg att ABAQUS inte har några förinställda enheter. 5.0 kan mätas i meter, fot eller vad som önskas. Det är upp till användaren att hålla reda på detta.
<code>rectangle</code>	Den nya ritytan har ett datafält <code>rectangle</code> som tilldelas hörn i origo och (4, 1.2).

Part()	En ny 2-dimensionell deformerbar del (part) skapas med namnet 'Part-1'. En "del" är en entitet som i fortsättningen kan ges form, materialegenskaper, finita elementnät med mera. Delen tilldelas en variabel p. Önskar användaren i fortsättningen anropa delen räcker det med att anropa p. Alternativt kan man använda den längre sökvägen <code>mdb.models['Model-1'].parts['Part-1']</code> .
BaseShell	Ritytans rektangel tilldelas den nya delen som därmed får en form.

Lösningar sökes till jämviktsekvationerna genom att skapa och köra ett arbete (job).

Exempel 2.4.2 Användaren har skapat modell, geometri, material, finita elementnät och randvillkor. Ett arbete skapas och körs med

```
myJob = mdb.models['Model-1'].Job(
    name='Job-1', model='Model-1',
    description='This is my model')
myJob.submit()
myJob.waitForCompletion()
```

Förklaring:

Job()	Skapar ett nytt arbete med namnet "Job-1" av modellen med namn "Model-1".
submit()	Arbetet skickas till ABAQUS-kärnan för analys. Lösning till jämviktsekvationerna sökes.
waitFor- Completion()	Ser till att exekveringen av skriptet gör uppehåll tills arbetet fullbordats.

Om ABAQUS finner lösningar till jämviktsekvationerna skapas en så kallad outputdatabas-fil (.odb) i temporärkatalogen. Odb-filen innehåller resultaten ordnade i så kallade bildrutor (frames) för varje steg (step). Första bildrutan innehåller information om kroppen i sitt odeformerade tillstånd. Övriga bildrutor innehåller information från resten av lösningsgången, till exempel förskjutningar och spänningar. En linjär analys ger såklart endast två bildrutor: kroppen i sitt odeformerade läge och

kroppen i sitt deformerade läge. Endast de resultat utskrivs som efterfrågas i output-modulen av preprocessen. Om däremot analysen av någon anledningen inte lyckas kan man felsöka i till exempel meddelandefilen (.msg). Där brukar man kunna hitta ledtrådar till vad som gått snett i analysen. Hur man får tillgång till resultaten i odb-filen framgår av Exempel 2.4.3.

Exempel 2.4.3 Användaren har skapat en modell med ett steg (step) som heter 'Step-1' och ett arbete som heter 'Job-1'. Arbetet sänds till kärnan, analysen är framgångsrik och resultaten återfinns i filen Job-1.odb. Efterfrågat är förskjutningarna i sista bildrutan för nod nummer 10.

```
odb = openOdb('Job-1.odb')
frame = odb.steps['Step-1'].frames[-1]
ux = frame.fieldOutputs['U'].values[9].data[0]
uy = frame.fieldOutputs['U'].values[9].data[1]
odb.close()
```

Förklaring:

<code>openOdb()</code>	Gör outputdatabasen tillgänglig.
<code>frames[-1]</code>	Informationen i sista bildrutan av 'Step-1' tilldelas en variabel <code>frame</code> för smidig åtkomst. Med Python fås sista elementet i en lista <code>L</code> som <code>L[-1]</code> .
<code>data[0]</code>	Förskjutningen i första frihetsgraden för 10:e noden skrivs till variabeln <code>ux</code> . Förskjutningar betecknas i ABAQUS med <code>U</code> .
<code>data[1]</code>	Förskjutningen i andra frihetsgraden för 10:e noden skrivs till variabeln <code>uy</code> .
<code>odb.close()</code>	Odb-filen stängs. Viktigt att stänga ner filen om användaren ämnar göra en ny simulering med samma arbetsnamn. Annars genereras ett felmeddelande.

I Exempel 2.4.4 visas hur användaren kan skriva data till en textfil.

Exempel 2.4.4 Fortsättning Exempel 2.4.3: användaren önskar skriva värdena på `ux` och `uy` till en textfil med namnet `datafile.txt`. Värdena avrundas till tre decimaler och avgränsas med indrag.

```
with open('datafile.txt', 'a') as outFile:
    outFile.write( str(round(ux,3))
    + '\t' + str(round(uy,3)) + '\n')
    outFile.close()
```

Förklaring:

<code>open()</code>	Öppna <code>datafile.txt</code> i tillfogningsläge (append mode). Det innebär att den befintliga textfilen (om den finns) ej raderas. Texten tillfogas endast.
<code>write()</code>	Värdena avrundas till tre decimaler, skrivs och avgränsas med indrag. Sist kommer ny rad <code>'\n'</code> . Notera hur det deklarerade blocket följer off-side-regeln.
<code>close()</code>	Stäng textfilen för redigering. Eftersom filen öppnades med <code>with</code> är denna rad i och för sig inte kritisk.

Det fina med skript-implementering är möjligheten att göra parameterstudier. Tag som exempel en användare som vill undersöka hur böjspänning i en platta för vissa fixa värden på längden beror på kvoten mellan bredd och längd. Användaren kan låta längd och bredd vara parametrar i pre-processen och omsluta pre- och postprocessen med en for-loop som löper över utvalda värden på kvoten. I varje iterationssteg analyseras modellen av kärnan, odb-filen öppnas och skrivs de efterfrågade spänningarna till en textfil som efter for-loopen är klar kan öppnas i till exempel Matlab för vidare analys. Det finns även möjlighet att analysera data och göra plottar i postprocessen i ABAQUS.

3 Försöksplanering

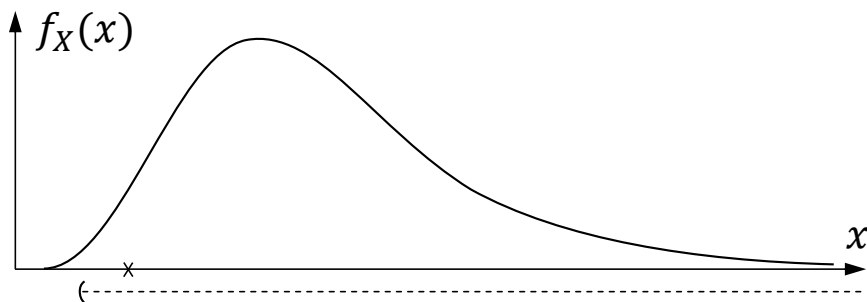
3.1 Allmänt

För att verifiera en beräkningsmodell genom provning av träkroppar finns det god vägledning i Boverkets handbok *Dimensionering genom provning* [8] och Sveriges Provnings- och forskningsinstitutets (SP) handbok *Dimensionering av träkonstruktioner genom provning* [47]. Skalan i modell och försök bör förhålla sig som 1:1. Materialet som provkroppen tillverkas av bör utväljas så, att hållfastheten inte väsentligen avviker från vad som förutsätts i beräkningsmodellen. Materialfuktighet och temperaturförhållanden ska övervakas eftersom träets hållfasthet tenderar att öka med avtagande fuktkvot och temperatur. I allmänhet uppmäts en högre bärförmåga för laster med kort varaktighet. Därför rekommenderar SP [47] att lasten anordnas så, att brott inträffar mellan 5 och 15 minuter efter pålastning. Krypningfenomen är vanskliga att verifiera i laboratorium, men praktiska metoder och råd finns i handböckerna. Ett exempel på detta är *Madisonkurvan* som uppskattar bärförmåga som funktion av lastvaraktighet. Särskild uppmärksamhet bör ägnas det generella verknings sättet hos provkroppen under pålastningen, då det på grund av oförutsedda händelser kan inträffa att verknings sättet avviker från det förväntade. Sådana avvikelser bör dokumenteras. Rande effekter på grund av provningsanordningen bör beaktas så att det som provas i laboratoriet verkligen överensstämmer med vad som beskrivs i beräkningsmodellen. Störst betydelse för bärförmågan har lastvaraktigheten och materialfuktigheten [47]. När försöksresultaten är sammanställda kan ett eventuellt samband med beräkningsmodellen eftersökas, ofta medelst en proportionalitetskonstant.

Enligt Boverkets handbok [8] och den numera upphävda europeiska normen ENV 1991-1 [17] bestäms det karakteristiska värdet av bärförmågan som 5-percentilen av bärförmågan med 75 % konfidensnivå. Innebörden av detta illustreras med en principskiss i Figur 3.1.

3.2 Dragmaskinen

Dragmaskinen är av märket Instron och består av en stålram med en 100 kN lastcell, se Figur 3.4 för en illustration. Provkroppen infästs mellan räfflade stålplattor genom friktion- och tryckverkan. Plattorna har ytan $50 \times 50 \text{ mm}^2$ och kan klämma om provbitar som är högst 10 mm breda. Provanordningen medger högst ca 70 cm långa provkroppar inklusive längden innanför infästningsplåtarna. Den ena infästningen är



Figur 3.1: Givet är en slumpfördelning med täthetsfunktion $f_X(x)$. Skattningen av 5-percentilen markeras med kryss. Ett nedåt begränsat konfidensintervall med konfidensgrad 75 % för skattningen markeras med streckad linje. Skattningen av 5-percentilen med 75 % konfidensgrad ges därmed av punkten markerad med parentes “(”.

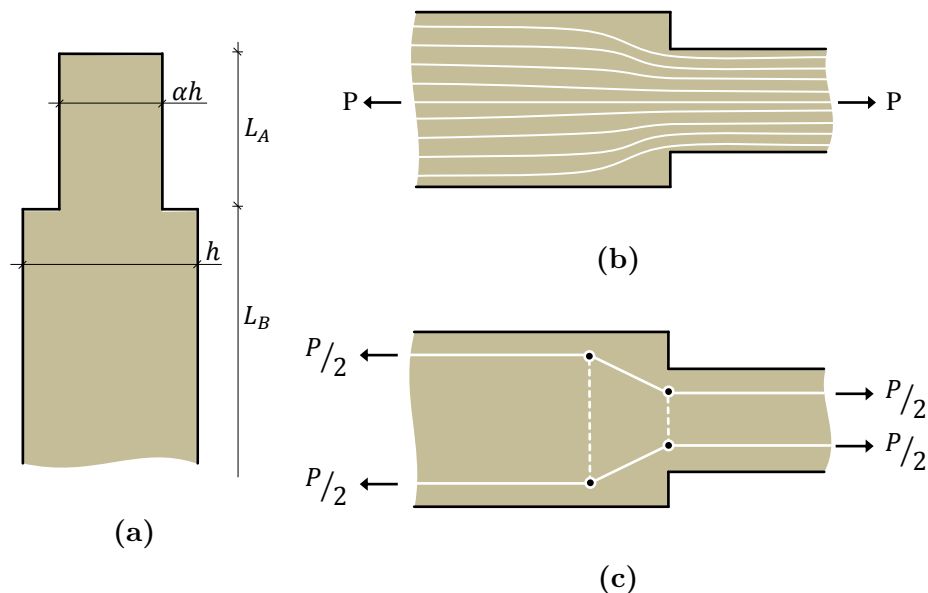
ledad med kula, den andra är oledad. Förstärkaren till lastcellen är en Hottinger Baldwin Messtechnik av modell MGC vilken mäter kraften med två samplingsar per sekund. Dragastigheten är 5 mm/min. Till förstärkaren ansluts en insamlingsdator.

3.3 Dimensionering av provkroppar

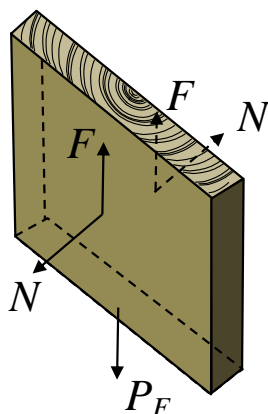
Tre grupper av provkroppar dimensioneras. Inom varje grupp har provkropparna lika djupa urtag på båda sidor om längsaxeln. Mellan grupperna varierar α mellan 0.3, 0.5 och 0.7 medan övriga geometriparametrar hålls konstanta. Geometrin efterliknar skissen i Figur 3.2a. Dimensionering av provkropparna görs med hänsyn till lastspridningseffekten bakom urtagen och med hänsyn till infästningsanordningen.

3.3.1 Dimensionering med hänsyn till lastspridningseffekten

Förberedande studier med FEM visar att det med hänsyn till lastspridningen bakom urtaget är nödvändigt med en längd om cirka fyra gånger tvärsnittshöjden. Lastspridningen framför urtaget kräver en längd om cirka en och en halv gånger tvärsnittshöjden. Lastspridningseffekten illustreras med ett tänkt spänningsfält i Figur 3.2b. Hur detta spänningsfält leder till transversell dragspänning (tvärdragspänning) illustreras med en så kallad strut-and-tie-modell i Figur 3.2c.



Figur 3.2: a) Provkropparna efterliknar schematiskt geometrin i figuren. Urtagen är lika stora på båda sidor om längsaxeln då α varierar mellan 0.30, 0.50 och 0.70. b) Förberedande studier med FEM visar att det med hänsyn till lastspridningen är nödvändigt med en längd om fyra gånger tvärsnittshöjden bakom urtaget. Ett tänkt spänningsfält illustreras. c) Med en strut-and-tie-modell kan det förstås hur tvärdrag (streckad linje) uppstår på grund av spänningsfältet i delfigur b.



Figur 3.3: Frilagd träbit mellan stålplattor. P_F balanseras av friktionskrafterna $2F$.

3.3.2 Dimensionering med hänsyn till infästningsanordningen

Provkropparna infästs i varje kortända med räfflade stålplattor som genom tryck- och friktionsverkan klämmer fast provbitarna. Dimensioneringen av provkropparna syftar till att uppfylla följande villkor:

1. provkropparna glider ej ur stålplattorna
2. provkropparna krossas ej mellan stålplattorna
3. provkropparna skjuvas ej sönder mellan stålplattorna
4. provkropparna går ej till dragbrott
5. provkropparna spricker vid urtagen

Genom att frilägga den del av provkroppen som befinner sig mellan stålplattorna och studera jämviktsekvationerna kan villkoren överslagsmässigt kvantifieras. Låt med beteckningen i Figur 3.3 P_F vara dragkraften som balanserar friktionskrafterna som utvecklas mellan tråkroppen och stålplattorna och ansätt ett rimligt värde på den statiska friktionskoefficienten $\mu = 0.8$. Eftersom stålplattorna har ytan $50 \times 50 \text{ mm}^2$ omformuleras villkoren således

1. $P_F < 2\mu f_{c90} h \alpha \cdot 0.05$
2. samma som 1.
3. $P_F < 2\tau h \alpha \cdot 0.05$
4. $P_F < f_{Lt} b h \alpha$
5. $P_F > \sqrt{2G_c E b^2 h \frac{\alpha}{1-\alpha}}$ enligt uppsprickningsvillkoret (4.8)

där

f_c	=	3.7 MPa	tryckhållfastheten vinkelrätt fibrerna
τ	=	5.3 MPa	skjuvhållfastheten
f_{Lt}	=	63 MPa	draghållfastheten parallellt fibrerna
E	=	9040 MPa	elasticitetsmodulen

Värden på materialparametrarna hämtas från [3]. I förekommande fall används medelvärdet av hållfastheten i radiell och tangentiell riktning.

Tabell 3.1: En överslagsmässig dimensionering av provkropparna leder till värdena enligt tabellen. Då har hänsyn tagits till lastspridningseffekten och infästningsanordningen.

	Provgrupp 1	Provgrupp 2	Provgrupp 3
α	0.3	0.5	0.7
b	~~~~~	6 mm	~~~~~
h	~~~~~	70 mm	~~~~~
L_A (framför)	~~~~~	minst 105 mm	~~~~~
L_B (bakom)	~~~~~	minst 280 mm	~~~~~

Värdet på G_c i Mod I antas förenklat 300 N/m och i Mod II 1000 N/m. Modblandningen utreds med FEM och noduppsläppningsmetod vilket visar att G_c håller sig nära nog konstant då α varierar mellan 0.3 och 0.7. Ett rimligt värde är $G_c = 600$ N/m, se avsnitt 6.1.4 för en förklaring.

Valda dimensioner

Lämpliga dimensioner som uppfyller samtliga av villkoren ovan ges i Tabell 3.1. Notera att begränsningen av maximal provkroppslängd i kombination med vald tvärsnittshöjd och lastspridningseffekten medför att provkropparna inte kan utformas med en transversell symmetrilinje vid halva längden. Av detta skäl bestämdes det att tillverka provkroppar med urtag endast på ena kortsidan, för att inte kompromissa med avseende på lastspridningseffekten. På grund av att stålplattorna som griper tag i träbiten har en begränsad storlek om 50 mm var det emellertid nödvändigt att anordna små urtag på den andra kortsidan också. Dessa urtag är i alla fall mindre än urtagen på andra kortsidan.

3.4 Utförande av provkroppar

Tre grupper om sju stycken provkroppar tillverkas med dimensioner enligt Tabell 3.2. Träkropparna är som lameller och har från början sågats ur ett K-virke av gran med tvärsnitt 70x45 mm². Träet har omsorgsfullt utvalts och besiktigats med avseende på förekomst av kvistar och andra defekter. Det finns näst intill inga synbara defekter i de färdiga provkropparna. Speciellt finns inga uppenbara defekter i närheten av urtagens hörn. Urtagen har formats med en figursåg, sandpapper och en fil med två sorters graderade ytor. Det har av praktiska skäl inte varit möjligt att

tillverka sju exakt likadana geometrier i varje grupp. Avvikelserna har bestämts med ett skjutmått, se vidare Tabell 3.2. Ett fotografi av tre färdiga provkropparna återfinns i Figur 3.5. Provkropparna har i 20 dagar före provningen förvarats i jämvikt med cirka 20°C och 60 % RF. För att bestämma temperatur och luftfuktighet har en kombinerad hygrometer/termometer av märket Voltcraft använts.

3.5 Genomförande av provning

Vid genomförandet observerades urtagens hörn med avseende på sprickbildning. Så snart en spricka blev synlig för blotta ögat noterades lasten. Provkropparna fortsatte sedan att dras i möjlig mån så att ett tydligt brott inträffade. Provningarna genomfördes den 6/11-2013 i M-husets laboratorium på Lunds Tekniska Högskola.

Tabell 3.2: Tre grupper om sju stycken provkroppar tillverkas med dimensioner enligt tabellen. Tjockleken och tvärsnittshöjden har uppmätts med skjutmått. Värdet på α är bestämt vid urtagets hörn. Provkroppsnumreringen är gjord med hänsyn till provningsresultaten.

	Prov nr	Tjocklek b [mm]	Tvärsnittshöjd h [mm]	α	Längd $L_A + L_B$ [cm]
Grupp 1 $\alpha \approx 0.30$	1	5.6	69.9	0.30	12.5+59.5
	2	5.5	70.3	0.28	12.5+59.5
	3	5.9	70.3	0.32	12.5+59.5
	4	5.7	69.8	0.29	12.5+59.5
	5	5.3	69.8	0.30	12.5+59.5
	6	5.7	69.5	0.28	12.5+59.5
	7	5.6	69.4	0.30	12.5+59.5
Grupp 2 $\alpha \approx 0.50$	8	5.9	69.9	0.52	12.5+57.5
	9	5.3	68.4	0.52	12.5+57.5
	10	5.4	68.7	0.53	12.5+57.5
	11	5.7	68.8	0.52	12.5+57.5
	12	5.7	68.7	0.53	12.5+57.5
	19	6.3	69.2	0.53	12.5+57.5
	20	6.1	68.9	0.51	12.5+57.5
Grupp 3 $\alpha \approx 0.70$	13	5.9	69.5	0.65	12.5+59.5
	14	5.4	69.5	0.71	12.5+59.5
	15	5.4	69.4	0.67	12.5+59.5
	16	5.6	69.2	0.66	12.5+59.5
	17	5.1	69.8	0.68	12.5+59.5
	18	5.5	70.1	0.65	12.5+59.5
	21	5.7	70	0.70	12.5+59.5



Figur 3.4: Fotografi av dragmaskinen Instron som användes vid provningen.



Figur 3.5: Fotografi av tre provkroppar, en från varje huvudgrupp.

4 Modellantaganden

4.1 Beräkningsmodeller

4.1.1 Analytisk metod för bestämning av uppsprickningslast

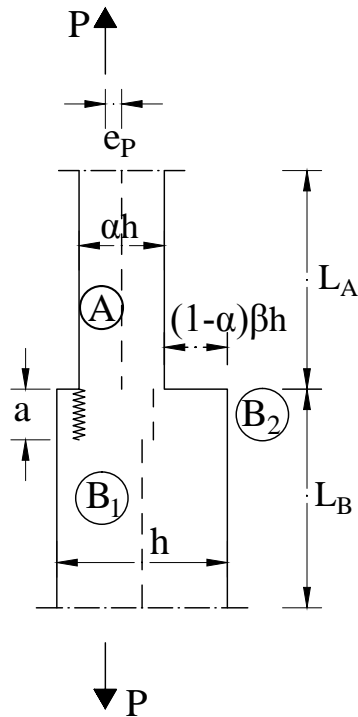
Fall 1 (en spricka)

I detta avsnitt härleds analytiska uttryck för den kritiska last som ger spricktillväxt i en dragbelastad kropp med urtag och en befintlig spricka. Det antas att materialet är linjärelastiskt och till grund ligger första ordningens balkteori enligt Bernoulli-Euler; det antas att plana tvärsnitt förblir plana och vinkelräta mot balkens medellinje, tvärsnitten deformeras inte, jämvikt gäller för det odeformerade läget samt töjningar och vinkeländringar är små så att givet vinkeländringen θ så är $\sin \theta \approx \theta$.

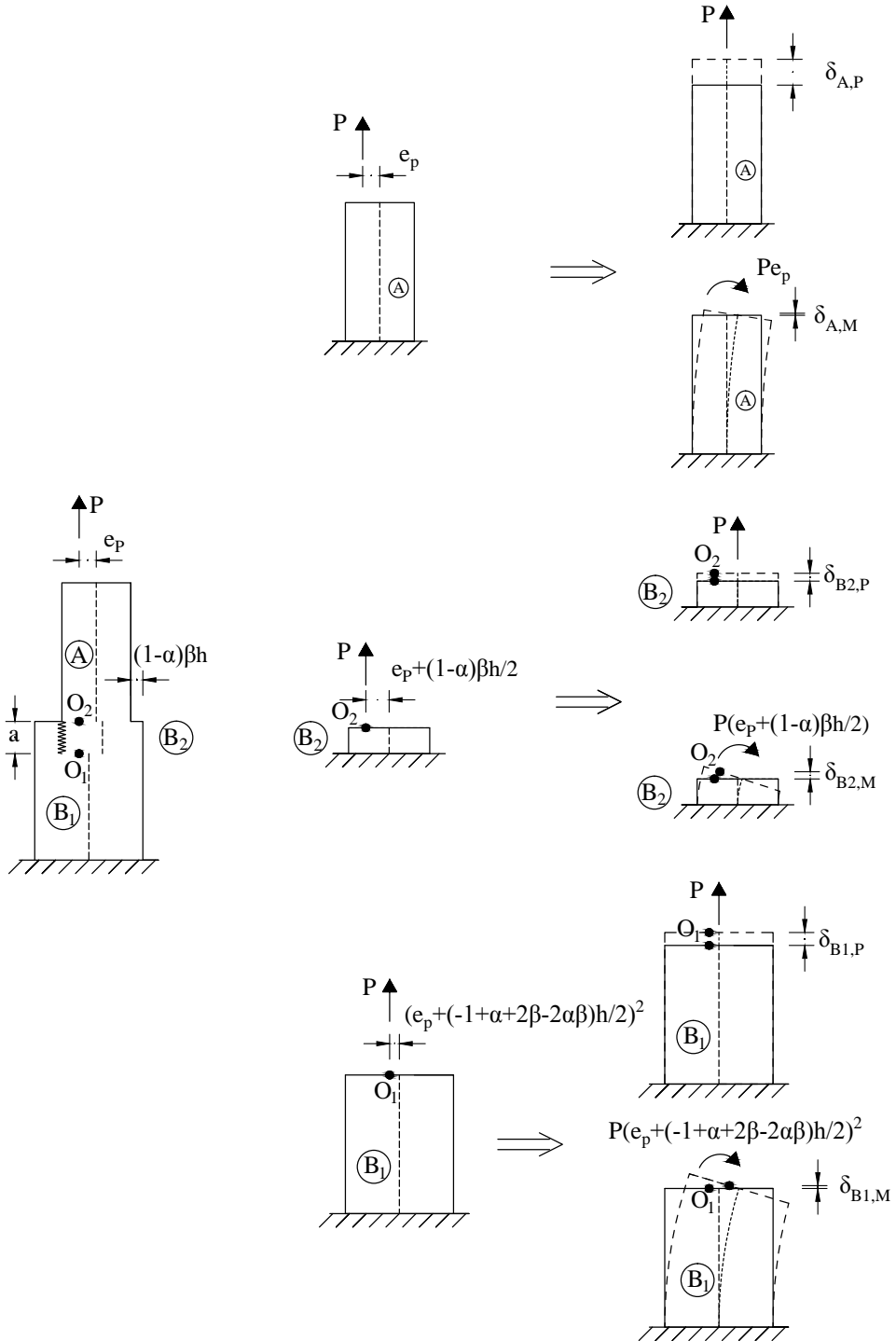
Betrakta en dragbelastad kropp med uppsprickning vid endast det ena urtaget så som i Figur 4.1. Den del av kroppen som är framför urtaget (se definition i Figur 1.2) benämns A och delen bakom urtaget benämns B . Längden på urtagsdelen betecknas L_A medan längden på delen bakom urtaget benämns L_B . Del B :s tvärsnittshöjd kallas h , och del A :s tvärsnittshöjd betecknas αh , där $0 < \alpha < 1$. Hela kroppen har tvärsnittsbredden b . I figuren är lastens excentricitet med avseende på del A :s centrumlinje betecknad e_P . Antag att uppsprickning sker på en viss sida om del A :s centrumlinje. Då betecknas urtagsdjupet på den motsatta sidan $(1 - \alpha)\beta h$. Det innebär att $0 < \beta < 1$ och att för $\beta = 0.5$ är de två urtagen lika djupa. När en spricka med längden a bildas får det främre området av del B , som befinner sig i höjd med sprickan, den nya tvärsnittshöjden $(\alpha + (1 - \alpha)\beta)h$. Det främre området på del B benämns nu del B_2 och det bakre benämns B_1 .

Kroppens tre delar A , B_1 och B_2 studeras var för sig. Den excentriska normalkraften ersätts med en statiskt ekvivalent centrisk normalkraft och ett fritt moment. I Figur 4.2 illustreras hur den totala förskjutningen δ_{tot} av kraftangreppspunkten kan bestämmas genom uppdelning i sex delar och summering av delarnas bidrag enligt superpositionsprincipen. För en fast inspänd och dragbelastad kropp med konstant rektangulärt tvärsnitt beräknas förskjutningen av kraftangreppspunkten med ekvationen

$$\delta_P = \frac{PL}{bhE} \quad (4.1)$$



Figur 4.1: Frilagd kropp med lastexcentriciteten e_P och med två urtag vars djup inte behöver vara lika. Området framför urtaget benämns A och har längden L_A . Området mellan urtag och sprickspets benämns B_2 och området bakom sprickspetsen benämns B_1 . Den totala längden av del B_1 och B_2 kallas L_B . Sprickan har längden a . Delarnas centrumlinjer är streckade. Tvärsnittshöjden bakom urtagen är h och framför urtagen αh . β beskriver urtagsdjupens andel på den sida av del A :s centrumlinje som är motsatt sprickan.



Figur 4.2: Dragbelastad konsol med urtag. Förskjutningen av kraftangreppspunkten bestäms genom uppdelning och summering enligt superpositionsprincipen.

där

P	centrisk normalkraft.
L	kroppens längd.
b	tvärsnittets bredd.
h	tvärsnittets höjd.
E	materialets elasticitetsmodul, det vill säga E_L (jämför Kapitel 2).

En konsol som utsätts för ett fritt moment M får vinkeländringen $\theta = \frac{ML}{EI}$ där I är tröghetsmomentet. Eftersom $M = Pe_P$ och $\sin \theta \approx \theta$ så blir förskjutningen på grund av momentet

$$\delta_M = \frac{PLe_P^2}{EI} \quad (4.2)$$

För del A blir förskjutningen av kraftangreppspunkten $\delta_A = \delta_{A,P} + \delta_{A,M}$ och (4.1) ger

$$\delta_{A,P} = \frac{PL_A}{b\alpha hE}$$

(4.2) ger

$$\delta_{A,M} = \frac{PL_A e_P^2}{EI_A}$$

För del B_2 blir förskjutningen på samma sätt

$$\delta_{B_2} = \delta_{B_2,P} + \delta_{B_2,M} = \frac{Pa}{b(\alpha + (1 - \alpha)\beta)hE} + \frac{Pa(e_P + (1 - \alpha)\beta h/2)^2}{EI_{B_2}}$$

och för del B_1

$$\begin{aligned} \delta_{B_1} &= \delta_{B_1,P} + \delta_{B_1,M} = \frac{P(L_B - a)}{bhE} + \dots \\ &\dots + \frac{P(e_P + (-1 + \alpha + 2(1 - \alpha)\beta)h/2)^2(L_B - a)}{EI_{B_1}} \end{aligned}$$

Tröghetsmomenten är

$$\begin{aligned} I_A &= \frac{b\alpha^3 h^3}{12} \\ I_{B_2} &= \frac{b(\alpha + (1 - \alpha)\beta)^3 h^3}{12} \\ I_B &= \frac{bh^3}{12} \end{aligned}$$

Den totala förskjutningen δ_{tot} fås genom summering av bidragen

$$\begin{aligned} \delta_{tot} = & \frac{P}{bhE} \left(\frac{L_A}{\alpha} + \frac{L_B - a}{1} \frac{a}{\alpha + \beta - \alpha\beta} + \dots \right. \\ & \dots + \frac{12e_p^2 L_A}{h^2 \alpha^3} + \frac{12a(e_p + (\beta - \alpha\beta)h/2)^2}{h^2(\alpha + \beta - \alpha\beta)^3} + \dots \\ & \left. \dots + \frac{12(e_p + (-1 + \alpha + 2\beta - 2\alpha\beta)h/2)^2(L_B - a)}{h^2} \right) \end{aligned} \quad (4.3)$$

Uttrycket i (4.3) insätts i brottvillkoret (2.3) sidan 22 (jämför Exempel 2.3.1) och deriveras med avseende på a vilket ger

$$\begin{aligned} \frac{P_c^2}{2bhE} \left(-1 + \frac{1}{\alpha + \beta - \alpha\beta} + \frac{12(e_p + (\beta - \alpha\beta)h/2)^2}{h^2(\alpha + \beta - \alpha\beta)^3} + \dots \right. \\ \left. \dots - \frac{12(e_p + (-1 + \alpha + 2\beta - 2\alpha\beta)h/2)^2}{h^2} \right) = G_c b \end{aligned} \quad (4.4)$$

Spricklasten P_c blir då

$$P_c = \frac{b\sqrt{2hG_cE}}{\sqrt{-1 + \frac{1}{\alpha + \beta - \alpha\beta} + \frac{12(e_p + (\beta - \alpha\beta)h/2)^2}{h^2(\alpha + \beta - \alpha\beta)^3} - \frac{12(e_p + (-1 + \alpha + 2\beta - 2\alpha\beta)h/2)^2}{h^2}}} \quad (4.5)$$

Bärförmågan är alltså en funktion av $b, h, G_c, E, \alpha, \beta$ och e_p . Varken spricklängden a eller L_A och L_B påverkar bärförmågan. För det förenklade fallet då urtagen är lika stora ($\beta = 0.5$) och lasten är centrisk ($e_p = 0$) kan (4.5) skrivas

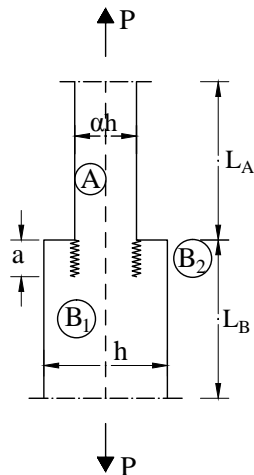
$$P_c = \frac{b\sqrt{2hG_cE}}{\sqrt{-1 + \frac{2}{1+\alpha} + 6\frac{(1-\alpha)^2}{(1+\alpha)^3}}} \quad (4.6)$$

Om urtaget är ensidigt ($\beta = 0$) och lasten centrisk blir (4.5)

$$P_c = \frac{b\sqrt{2hG_cE}}{\sqrt{-1 + \frac{1}{\alpha} - 3(1-\alpha)^2}} \quad (4.7)$$

Fall 2 (två sprickor)

Betrakta nu en dragbelstad kropp för vilken uppsprickning sker simultant vid de två urtagen. Det antas att kroppen har en symmetrilinje parallellt med sin längsaxel och att belastningen är centrisk. Detta innebär med



Figur 4.3: Frilagd kropp med två likadana sprickor med längden a och lika djupa urtag.

beteckningarna i Figur 4.3 att $\beta = 0.5$ och $e_P = 0$. Vid urtagens hörn finns lika långa sprickor med längden a . I detta fall innebär uppsprickning att det främre området av del B , benämnt del B_2 , får samma tvärsnittsegenskaper som del A . Med del B_1 avses del B utan del B_2 . Förskjutningen av kraftangreppspunkten ges därför av de tre bidragen

$$\begin{aligned}\delta_A &= \frac{PL_A}{b\alpha hE} \\ \delta_{B_2} &= \frac{Pa}{b\alpha hE} \\ \delta_{B_1} &= \frac{P(L_B - a)}{bhE}\end{aligned}$$

Då är kroppens totala förlängning på grund av kraften P

$$\delta = \frac{P}{bhE} \left(\frac{L_A}{\alpha} + \frac{a}{\alpha} + \frac{L_B - a}{1} \right)$$

och insättning i (2.3), jämför Exempel 2.3.1, ger uppsprickningsvillkoret

$$\frac{P_c^2}{2bhE} \left(\frac{1 - \alpha}{\alpha} \right) = 2G_c b$$

Faktorn 2 i högerledet är en följd av att två sprickor propagerar simultant. Villkoret för spricklasten blir

$$P_c = \frac{b\sqrt{4hG_cE}}{\sqrt{(1 - \alpha)/\alpha}} \quad (4.8)$$

Tabell 4.1: Kritisk tøjningsenergi som används i simuleringen. Värdena är hämtade från [46].

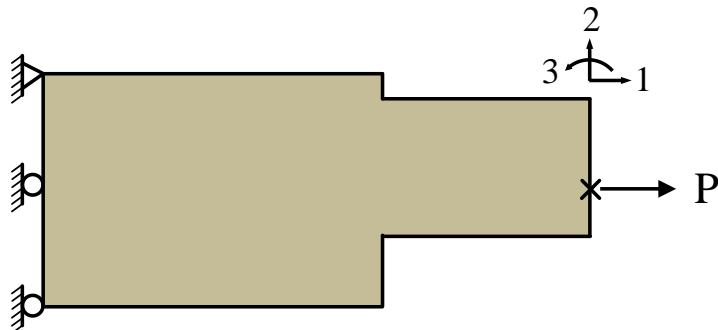
	Mod I	Mod II
Kritisk tøjningsenergi [Nm/m ²]	283	1240

vilket kan jämföras med (4.7) som ger P_c för samma geometri, men med uppsprickning bara vid ena urtaget. Bärförmågan är i detta fall en funktion av b, h, G_c, E och α . Varken spricklängden a eller L_A och L_B påverkar bärförmågan.

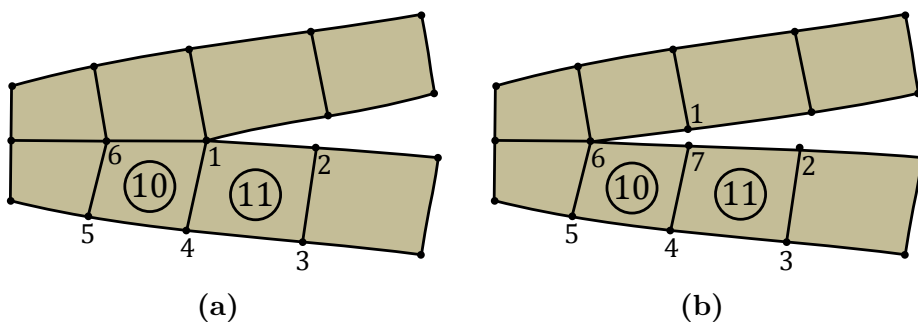
4.1.2 FEM med noduppsläppning

En numerisk beräkningsmodell med FEM och noduppsläppning används för att simulera verkningssättet hos en dragbelastad träkropp. Se Bilaga 2 för en förklaring av FEM. Modellen bygger på första ordningens linjära elasticitetsteori och jämviktsekvationerna löses med ABAQUS. Tøjningarna är små och geometrisk ickelinjäritet försummas. De finita elementen utgörs av isoparametriska 4-nodsskivelement (CPS4R) i plant spänningstillstånd med reducerad integration och timglaskontroll. Elementen som är nära nog kvadratiska har en sidlängd som är ungefär 2 % av tvärsnittshöjden, se Bilaga 5 för en illustration av elementnätet. Denna sidlängd är uppskattningsvis lämplig med hänsyn till konvergens och krav på effektivitet och har bestämts med parameterstudier. Materialmodellen är homogent ortotrop med konstitutivt samband enligt (2.1), sidan 16 och kritiska tøjningsenergi i Mod I och Mod II enligt Tabell 4.1. Randvillkoren innebär att kroppen är fastspänd i sin ena kortsida, se Figur 4.4 för en illustration och en definition av frihetsgraderna. Noderna längs kortsidan är förhindrade att röra sig parallellt längdriktningen men endast en nod är förhindrad att röra sig transversellt längdriktningen. Noderna längs andra kortsidan är i första och tredje frihetsgraderna kinematiskt kopplade till en referenspunkt på vilken en punktlast verkar. På detta sätt är det behändigt att simulera en excentrisk belastning: endast referenspunkten behöver förflyttas.

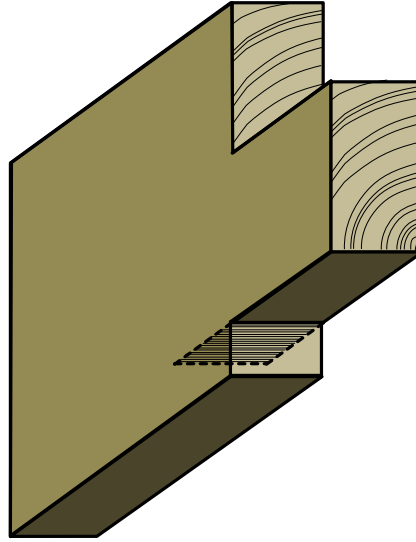
Genom att släppa upp noder längs en tänkt brottyta går det att uppskatta mängden frigjord tøjningsenergi ΔU_e . Låt nämligen $U_{e,i}$ vara systemets tøjningsenergi när nod i släpps upp. Då är $\Delta U_e = U_{e,i+1} - U_{e,i}$. Den tänkta sprickan börjar vid urtagets hörn och löper parallellt längdriktningen. För att ΔU_e ska vara ungefärligen konstant när nod



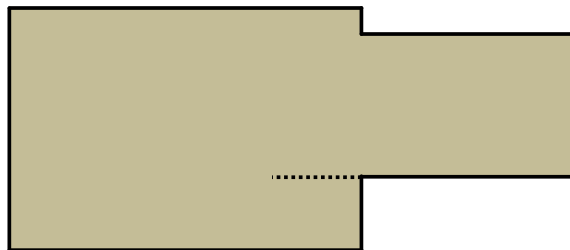
Figur 4.4: Randvillkoren innebär att noderna längs ena kortsidan är förhindrade att röra sig parallellt längdriktningen medan endast en nod är förhindrad att röra sig transversellt längdriktningen. Noderna längs andra kortsidan är i första och tredje frihetsgraderna kinematiskt kopplade till en referenspunkt på vilken verkar en punktlast. Referenspunkten är i figuren markerad med ett kryss. Frihetsgradernas definition framgår av figuren.



Figur 4.5: Principen vid noduppsläppning illustreras här i två delfigurer. Betrakta element nr 10 och 11 som båda kopplar till nod nr 1 i delfigur a). Genom att lägga till en ny nod, nr 7, och koppla elementen till denna istället, såsom i delfigur b), säges nod nr 1 ha släppts upp.



(a)



(b)

Figur 4.6: Eftersom simuleringen görs i plant spänningstillstånd kommer den verkliga brottytan (yta begränsad av streckade linjer i delfigur a) att projiceras på en linje (streckad linje i delfigur b). Därför är "riktningen" som brottytan utbreder sig längs väldefinierad. Mod I och Mod II-förskjutningar blir också väldefinierade.

nummer i släpps upp jämfört med när nod nummer $i + 1$ släpps upp (två på varandra följande noder), så fordras en initiell spricka motsvarande att ungefär 20 noder släppts upp. Noduppsläppningen åstadkoms genom att redigera den så kallade input-filen i ABAQUS. En ny nod läggs till med samma koordinater som den befintliga noden vid sprickspetsen. Två av de fyra element som kopplar till sprickspetsnoden redigeras så att de istället kopplar till den nya noden. Jämför med Figur 4.5a och b för en illustration över förfarandet. Spricklasten bedöms genom att först beräkna en utnyttjandegrad som definieras enligt

$$R = \sqrt{\frac{\Delta U_e}{G_c \Delta A}}$$

där

R	Utnyttjandegraden
ΔU_e	Ändring i elastisk töjningsenergi
G_c	Kritisk töjningsenergi i det aktuella fallet
ΔA	Frigjord brottyta beräknad som den finita elementlängden multiplicerat med tjockleken

Om till exempel utnyttjandegraden för en kropp belastad med P bestäms som R , kan spricklasten beräknas med $P_c = \frac{P}{R}$. Denna beräkning fungerar enligt författarnas erfarenhet väl och orsaken är det linjärelastiska modellantagandet.

Värdet på G_c beror i det aktuella fallet på modblandningen vilken bestäms med utgångspunkt i den relativa förskjutningen mellan de båda noderna närmast sprickspetsen. Kalla den relativa förskjutningen transversellt riktningen för brottytans utbredning för δ_I och motsvarande förskjutning parallellt för δ_{II} . Eftersom modellering görs i 2D så är riktningen för brottytans utbredning väldefinierad, jämför Figur 4.6a och b. Andelen Mod I beräknas nu som

$$r_I = \frac{\delta_I}{\delta_I + \delta_{II}} \quad (4.9)$$

och andelen Mod II på analogt vis. Vidare antas linjär interaktion vid beräkning av G_c från $G_{c,I}$ och $G_{c,II}$ vilket ger

$$G_c = r_I G_{c,I} + r_{II} G_{c,II} \quad (4.10)$$

där

Tabell 4.2: Hållfasthet och brottenergi som används i simuleringen. Värdena är hämtade från [46].

	Mod I	Mod II
Hållfasthet [MPa]	2.9	9.8
Brottenergi [Nm/m ²]	283	1240

G_c	Kritisk töjningsenergi i det aktuella fallet
r_I	Andelen Mod I i modblandningen
r_{II}	Andelen Mod II i modblandningen
$G_{c,I}$	Kritisk töjningsenergi för ren Mod I
$G_{c,II}$	Kritisk töjningsenergi för ren Mod II

4.1.3 FEM med kohesiv zonmodellering

Med kohesiv zonmodellering antas en fördefinierad potentiell brottyta som begynner i ena urtagets hörn och löper parallellt med längdriktningen till motstående kortsida. Brottzonens verknings sätt simuleras med plasticitetsteori. Utanför brottzonen är materialet elastiskt och ortotrop. Töjningarna antas små och geometrisk ickelinjäritet försummas. Huvudsakligen används isoparametriska 4-nodsskivelement (CPS4R) i plant spänningstillstånd med reducerad integration och timglaskontroll. I mindre omfattning förekommer 3-nodsskivelement (CPS3R) i plant spänningstillstånd. 4-nodselementen definieras med en sidlängd som är 3.5 % av tvärsnittshöjden, se Bilaga 5 för en illustration av elementnätet. Denna sidlängd är uppskattningsvis lämplig med hänsyn till konvergens och krav på effektivitet och har bestämts med parameterstudier. Det finita elementnätet består av två huvuddelar vilka tangerar varandra längs med den kohesiva zonen. Nätet är anordnat så, att huvuddelarnas noder överlappar varandra i par längs med den kohesiva zonen. Mellan varje nodpar assembleras två olinjära fjädrar, den ena för att svara mot Mod I-förskjutning och den andra mot Mod II, se Figur 4.7a för en illustration av den beskrivna brottzonen och fjäderparen. Fjädrarna är okopplade och deras konstitutiva samband beskrivs i Figur 4.7b. Vilka värden som används i materialmodellen framgår av Tabell 4.2. Randvillkoren innebär att kroppen är fastspänd i sin ena kortsida. Noderna längs kortsidan är förhindrade att röra sig parallellt längdriktningen. Tre noder (bredvid varandra på rad) låses i riktning transversellt längdriktningen. Det går enligt författarnas erfarenhet inte att få stabilitet i lösningen

av jämviktsekvationerna om endast en nod låses i transversell riktning. Denna egendomlighet tillskrivs mjukvarans sätt att hantera den olinjära responsen. Noderna längs andra kortsidan är i första och tredje frihetsgraderna kinematiskt kopplade till en referenspunkt. Denna är i sin tur kopplad till en förskjutningsstyrd last. På detta sätt är det behändigt att simulera en excentrisk belastning; endast referenspunkten behöver förflyttas. Illustrationen i Figur 4.4 är i princip tillämplig för randvillkoren i det här fallet, med undantag av att antalet noder som låses i riktning transversellt ländriktningen är tre samt att belastningen är förskjutningsstyrd. Definitionen av frihetsgraderna är densamma. Dessutom införs ett kontaktvillkor som innebär att elementen på båda sidor om den kohesiva zonen inte överlappar varandra. Detta åstadkoms i ABAQUS med ett så kallat "hårt" kontaktvillkor med penalty-formulering. På grund av fjädrarnas bilinjära konstitutiva samband erhålls en olinjär global respons. Lösning bestäms stegvis med Newton-Raphsons metod.

Spricklasten utvärderas genom att töjningsenergierna i respektive mod kombineras i brottvillkoret

$$\left(\frac{G_I}{G_{f,I}}\right)^n + \left(\frac{G_{II}}{G_{f,II}}\right)^n \leq 1$$

där

G_I	Frigjord töjningsenergi Mod I-fjäder
G_{II}	Frigjord töjningsenergi Mod II-fjäder
$G_{c,I}$	Brottenergi Mod I
$G_{c,II}$	Brottenergi Mod II
n	Exponent som antas vara 1

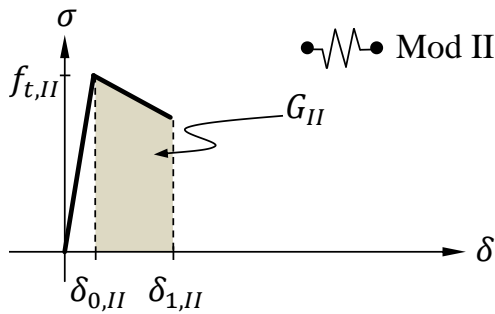
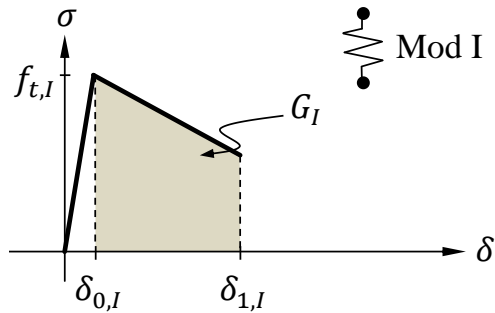
Det antas alltså att töjningsenergierna i Mod I och Mod II interagerar i princip enligt Figur 4.7c. Töjningsenergierna bestäms i varje beräkningssteg genom integration av den olinjära delen av fjädrarnas spännings-förskjutningskurva.

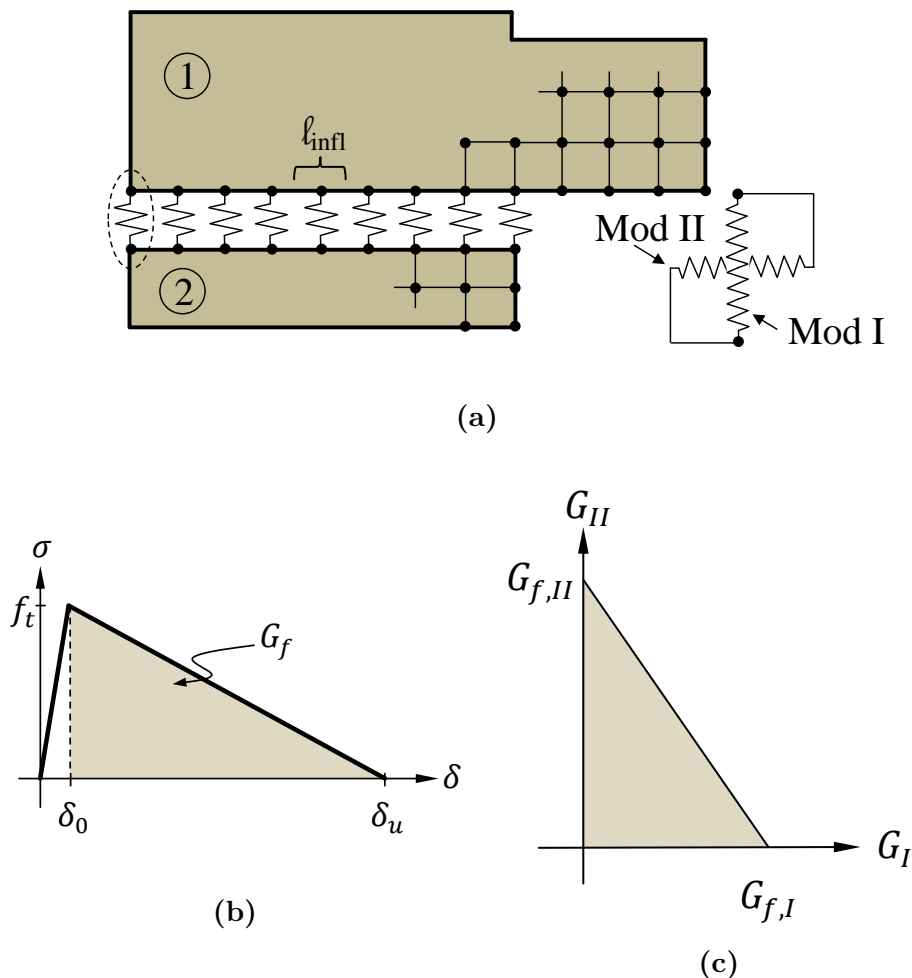
Exempel 4.1.1 I ett visst beräkningssteg och för ett visst fjäderpar längs med kohesiva randen gäller att Mod I-fjäderspar har förskjutningen $\delta_{1,I}$ och Mod II-fjäderspar $\delta_{1,II}$. Töjningsenergierna G_I och G_{II} bestäms genom att beräkna ytan under spännings-förskjutningskurvorna såsom illustreras i figurerna nedan. Om det

gäller att

$$\frac{G_I}{G_{f,I}} + \frac{G_{II}}{G_{f,II}} > 1$$

så antas att brott har uppstått i fjäderparet. I annat fall antas fjäderparet inte ha gått till brott. Varje fjäderpar längs med den kohesiva randen får på så vis kontrolleras i varje beräkningssteg, eftersom det inte är säkert exakt var det mest belastade paret finns.





Figur 4.7: a) Med kohesiv zonmodellering antas en fördefinierad potentiell brottyta. Det finita elementnätet består av två delar, betecknade med (1) och (2) i figuren. Nätet är utformat så, att två noder möts i par längs med den kohesiva zonen. Mellan varje nodpar assembleras två fjädrar vilka svarar mot Mod I och II-förskjutning. Influensytan för en fjäder är med figurens beteckning l_{infl} multiplicerat med tjockleken. Inom streckade linjer syns ett nodpar med fjädrar som får halva influensytan. b) Det konstitutiva sambandet för en fjäder är bilinjärt. Vid förskjutningen δ_0 upphör det elastiska beteendet. Vid δ_u upphör fjädern att bära last och går till brott. δ_0 kan i praktiken inte väljas hur litet som helst, eftersom lösningen av jämviktsekvationerna saknas i så fall. I simuleringen sätts $\delta_0 = \delta_u \cdot 10^{-3}$. Värdet på δ_u ges av att ytan under grafen till höger om δ_0 är lika med brottenergin G_f för den aktuella moden. f_t är det ortotropa materialets hållfasthet i den aktuella moden. c) Brottvillkoret bygger på att töjningsenergierna samspelar linjärt enligt figuren.

4.2 Slumpmodell

De analytiska och numeriska beräkningsmodellerna verifieras genom provning av små träkroppar. På grund av att trä inte är ett exakt homogent och ortotropt material kommer observerade bärförmågor att variera. Dessutom är det svårt att helt utesluta randeffekter i infästningsanordningen och att bestämma provkropparnas dimensioner utan fel. Mätutrustningen är också behäftad med viss osäkerhet. För att hantera den totala effekten av osäkerheterna används följande slumpmodell som är hämtad från Boverkets handbok [8]

$$P_i = DP_{t,i}(X_i) \cdot \epsilon_i \quad (4.11)$$

där

P_i	bärförmåga bestämd genom provning nr i
D	beräkningskoefficient som skall bestämmas genom provning
$P_{t,i}$	antagen teoretisk modell med variabeln X_i
X_i	deterministiska storheter
ϵ_i	slumpavvikelser sådana att för $i = 1, 2, 3, \dots$ så är $\ln \epsilon_i \in N(0, \sigma)$ och $\epsilon_1, \epsilon_2, \epsilon_3, \dots$ är oberoende stokastiska variabler

Med en transformation fås den linjära modellen

$$\ln P_i = \ln D + \ln P_{t,i}(X_i) + \ln \epsilon_i \quad (4.12)$$

Som skattning av $\ln D$ används medelvärdet

$$(\ln D)^* = \frac{1}{n} \sum_i \ln \frac{P_i}{P_{t,i}(X_i)} \quad (4.13)$$

som har medelfelet

$$d(\ln D)^* = \frac{s}{\sqrt{n}}$$

där s^2 är stickprovsvariansen

$$s^2 = \frac{1}{n-1} \sum_i \left(\ln \frac{P_i}{P_{t,i}(X_i)} - \frac{1}{n} \sum_j \ln \frac{P_j}{P_{t,j}(X_j)} \right)^2 \quad (4.14)$$

och av Gauss approximationsformler följer den approximativt väntevärdesriktiga skattningen D^* av D

$$D^* = e^{(\ln D)^*} \quad (4.15)$$

med den approximativa variansen

$$\text{Var} [(D^*)] = \text{Var} [(\ln D)^*] \cdot \left(e^{E[(\ln D)^*]} \right)^2$$

där $E[(\ln D)^*]$ betecknar väntevärdet av $(\ln D)^*$ [53].

Låt $I_{\ln D} = [(\ln D_-)^*, (\ln D_+)^*]$ vara en intervallskattning av $\ln D$ med konfidensnivå α . Då följer det av Gauss approximationsformler och det faktum att exponentialfunktionen är monoton att ett konfidensintervall med den ungefärliga konfidensgraden α för D är

$$I_D = [e^{(\ln D_-)^*_{obs}}, e^{(\ln D_+)^*_{obs}}]$$

där subskriptet $_{obs}$ betecknar det observerade värdet av den stokastiska variabeln.

5 Metod

5.1 Verifiering och kalibrering genom provning och simulering

5.1.1 Verifiering av beräkningsformel för två likadana urtag med centrisk belastning

Handberäkningsformeln verifieras genom provning för fallet med två likadana urtag och centrisk belastning. Det vill säga parametern α varieras medan $\beta = 0.5$ och $e_p = 0$. Verifieringen görs både med antagande om en spricka och med antagande om två simultana sprickor, det som motsvarar handberäkningsformel Fall 1 respektive Fall 2. Den valda metoden illustreras i Figur 5.1. För att beskriva slumpmässigheten i de uppmätta spricklasterna ansätts en slumpmodell enligt (4.11). Inför följande definition

Definition 5.1.1 Den skenbara brottsegheten är

$$\widetilde{K}_c = \sqrt{G_c E_L} \quad (5.1)$$

där G_c är den kritiska energifrigörelsen och E_L är elasticitetsmodulen i longitudinell riktning.

Med antagande om isotropt material överensstämmer den skenbara brottsegheten \widetilde{K}_c med den verkliga brottsegheten K_c . Så är inte fallet med antagande om ortotropt material.

Den skenbara brottsegheten \widetilde{K}_c i (4.6) och (4.8) lyfts ut ur de deterministiska storheterna $X_{t,i}$ i slumpmodellen (4.11). Skattningen av beräkningskoefficienten i slumpmodellen (4.11) blir en skattning av \widetilde{K}_c i

$$P_i = \widetilde{K}_c P_{t,i}(X_i) \cdot \epsilon_i \quad (5.2)$$

där

P_i	bärförmåga bestämd genom provning nr i
\widetilde{K}_c	skenbar brottseghet som bestäms genom provning
$P_{t,i}$	beräkningsuttryck med variabeln X_i
X_i	material- och geometriparametrar exklusive brottsegheten

ϵ_i slumpavvikelser sådana att för $i = 1, 2, 3, \dots$ så är $\ln \epsilon_i \in N(0, \sigma)$ och $\epsilon_1, \epsilon_2, \epsilon_3, \dots$ är oberoende stokastiska variabler

Det följer av (5.2) att \widetilde{K}_c antas vara konstant med α , åtminstone i det observerade intervallet $0.3 < \alpha < 0.7$. I en förstudie med FEM och noduppsläppningsmetod framgår det att andelen Mod I uppgår till 0.60, 0.59 och 0.58 för $\alpha = 0.3$, $\alpha = 0.5$ respektive $\alpha = 0.7$. Verifiering av antagandet att \widetilde{K}_c är konstant söks genom provning och statistiska metoder. Hypotesen att den skenbara brottsegheten är lika för alla tre provgrupper prövas på signifikansnivån 5 % med simultana konfidensintervall. Testets styrka beräknas i närheten av de observerade skillnaderna med hypotesen att den skenbara brottsegheten är lika mellan något par bland de tre provgrupperna. \widetilde{K}_c skattas med (4.15). Karakteristiskt värde härleds genom att bestämma ett nedåt begränsat konfidensintervall för 5-percentilen med 75 % konfidensnivå med Lawless intervallskattningsmetod för den stokastiska variabeln $(\ln D)^*$ i (4.13) som enligt antagande är normalfördelad och transformera enligt (4.15) med användande av Gauss approximationsformler [30]. Se vidare avsnitt 5.2.1. Slumpmodellens giltighet bedöms genom att 1) studera hur uppmätta värden på \widetilde{K}_c avviker från den observerade skattningen $(\widetilde{K}_c)^*$, 2) studera lognormalfördelningsplotten och 3) testa hypotesen att data kommer från en lognormalfördelning med ett Kolmogorov-Smirnov-test. Beräkningsformelns förmåga att förklara data bedöms genom att jämföra uppmätta värden med teoretiska då skattat värde på \widetilde{K}_c används. Jämförelsen görs med en linjär regressionsanalys där interceptet antas vara 0 och förklaringsgraden R^2 noteras. Förklaringsgraden definieras på samma sätt som kvadraten av korrelationskoefficienten för två stokastiska variabler, se vidare [6, 7].

5.1.2 Kalibrering av numeriska beräkningsmodeller genom provning

Uppsprickningslasten bestämd genom provning jämförs med datorsimuleringar. De simulerade provkropparna har två likadana urtag vars djup varieras mellan 0.3, 0.5 och 0.7. Simuleringarna görs med FEM och noduppsläppning respektive kohesiv zon. Slumpmässigheten i de uppmätta spricklasterna beskrivs av slumpmodellen (4.11). Proportionalitetsfaktorn D skattas med ett medelvärde enligt (4.15) och karakteristiskt värde härleds. Slumpmodellens giltighet bedöms genom att

1) studera hur uppmätta värden på $\ln D$ avviker från den observerade skattningen $(\ln D)^*$, 2) studera lognormalfördelningsplotten och 3) testa hypotesen att data kommer från en lognormalfördelning med ett Kolmogorov-Smirnov-test. De numeriska beräkningsmodellernas förmåga att förklara data bedöms och jämförs med varandra.

5.1.3 Bestämning av den skenbara brottsegheten \widetilde{K}_c genom simulering

Modblandningens α -, β - och e_p -beroende

För sex olika värden på β varieras α och e_p/h samtidigt på fem nivåer och modblandningens andel Mod I beräknas med FEM och noduppsläppning. Parametervärdena återfinns i Figur 5.2. För de fixa värdena $\alpha=0.5$, 0.6 och 0.7 anpassas krökta ytor som med hög korrelation beskriver β - och e_p -beroendet. Anpassningen görs i Matlab med anropet `fit` och parametern `'poly33'`. Metoden är olinjär multipel regression. De krökta ytorna beskrivs av polynom med sammanlagt tio parametrar. För $\alpha \geq 0.8$ ansätts förenklat andelen Mod I konstant lika med 0.6 för alla värden på β och e_p .

G_c som funktion av andelen Mod I

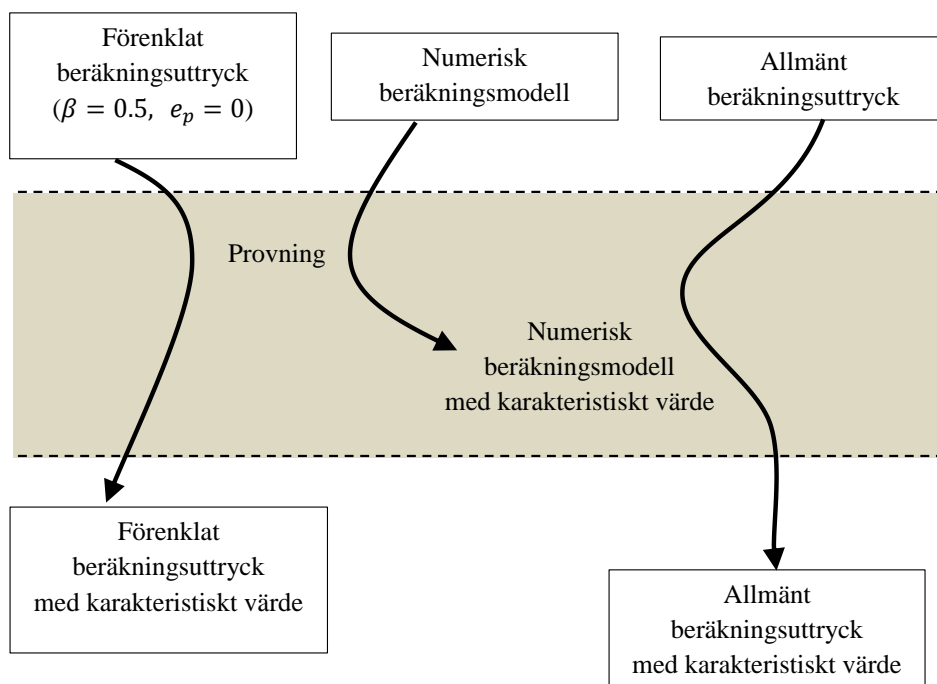
G_c beräknas som en linjärkombination av G_I och G_{II} . Med $G_I = 283$ och $G_{II} = 1240$ kan G_c uttryckas som

$$G_c = 283r_I + 1240r_{II} \quad (5.3)$$

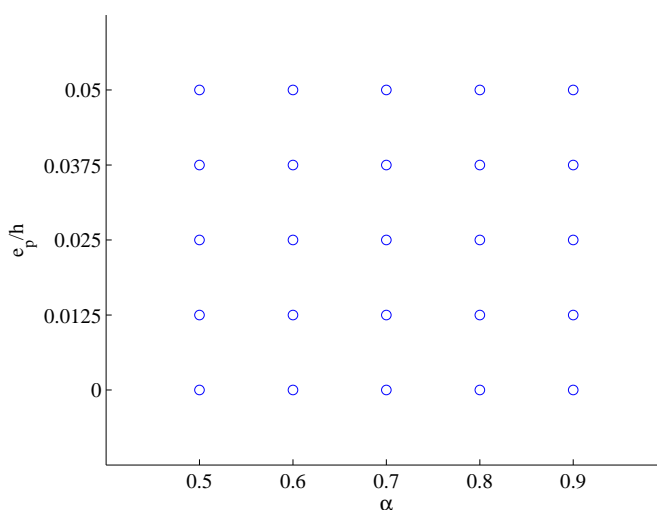
där r_I och r_{II} står för andelen Mod I respektive andelen Mod II i modblandningen och $r_I + r_{II} = 1$.

Bestämning av \widetilde{K}_c

Den skenbara brottsegheten \widetilde{K}_c vilken definieras i (5.1) ges direkt av beräknad andel Mod I och (5.3). Medelvärde på elasticitetsmoduler och skjuvmoduler återfinns i Tabell 2.2 sidan 15 och dessa används i simuleringarna.



Figur 5.1: Schemat ovan illustrerar metoden som används. De översta tre ruttorna symboliserar beräkningsmodellerna, i mitten syns verifieringsmetoderna och nederst visas resultaten.



Figur 5.2: Värden på α och e_p för bestämning av brottsegheten. β varierar mellan 0, 0.1, 0.2, 0.3, 0.5 och 0.75. Totalt görs 150 simuleringar.

5.1.4 Verifiering av beräkningsformel allmänna fallet

Handberäkningsformeln (4.5) sidan 52 (Fall 1 en spricka) ger med insättning av den skenbara brottsegheten (5.1) ett beräknat värde på bärförmågan för det allmänna fallet då $0.5 < \alpha < 1$ och $0 < \beta, e_P/h < 1$. Verifiering görs genom simulering av 150 provkroppar där α, β och e_P varieras enligt schemat i Figur 5.2. Simuleringsresultaten kalibreras med beräkningskoefficienten D_{obs}^* (se avsnitt 5.1.2). Karakteristisk bärförmåga bestäms med hjälp av 5-percentilen av D_{obs}^* . Den valda metoden illustreras i Figur 5.1.

5.2 Statistiska metoder

5.2.1 Konfidensintervall för normalpercentiler

Låt för $i = 1, \dots, n$: $X_i \in N(\mu, \sigma)$ vara oberoende och likafördelade stokastiska variabler och betrakta percentilen

$$\kappa_\alpha = \mu + z_\alpha \sigma \quad (5.4)$$

där z_α är 100α -percentilen i en $N(0, 1)$ -fördelning, $0 < \alpha < 1$ [10]. En intervallskattning av κ_α kan med Lawless metod härledas genom att 1) punktskatta κ_α med hjälp av Maximum-likelihood-metoden (ML-metoden) och 2) bilda en testvariabel med känd fördelning [30]. I ML-metoden skattas en parameter θ med tillhörande parameterområde A med det värde $\theta^* \in A$ som maximerar den så kallade Likelihoodfunktionen. Om utfallsrummet är kontinuerligt ges Likelihoodfunktionen $L(\theta)$ av

$$L(\theta) = \prod_i f(x_i; \theta)$$

där x_1, \dots, x_n är ett slumpmässigt stickprov från en fördelning med täthetsfunktion $f(x; \theta)$ [7].

En ML-skattning av κ_α är enligt Chakraborti & LiSource [10]

$$\hat{\kappa}_\alpha = \bar{x} + z_\alpha s \quad (5.5)$$

där medelfelet $s = \sqrt{\frac{1}{n-1} \sum (x_i - \bar{x})^2}$. Testvariabeln som Lawless använder är $U_\alpha = \frac{\hat{\kappa}_\alpha - \kappa_\alpha}{s}$. Betrakta uttrycket

$$P(U_\alpha \leq c) \quad (5.6)$$

och notera att (5.6) med hjälp av (5.5) kan omskrivas

$$P(\kappa_\alpha \geq \bar{x} + (z_\alpha - c)s) \quad (5.7)$$

vilket visar att κ_α kan intervallskattas med hjälp av testvariabeln U_α . Insättning av (5.4) och (5.5) i (5.6) ger

$$P\left(\frac{\bar{x} + z_\alpha s - (\mu + z_\alpha \sigma)}{s} \leq c\right)$$

som med en omskrivning blir

$$P(U_\alpha \leq c) = P\left(\frac{\sqrt{n}((\bar{x} - \mu)/\sigma) - \sqrt{n}z_\alpha}{s/\sigma} \leq (c - z_\alpha)\sqrt{n}\right) \quad (5.8)$$

Nu är $\sqrt{n}(\bar{x} - \mu)/\sigma \in N(0, 1)$ och $(n-1)s^2/\sigma^2 \in \chi_{(n-1)}^2$ oberoende stokastiska variabler vilket visas med bland annat Cochrans sats [7]. Därmed får uttrycket till vänster om olikheten i (5.8) en så kallad icke-centrisk t-fördelning $t'_{(n-1)}(-\sqrt{n}z_\alpha)$ med $(n-1)$ frihetsgrader och parameter $-\sqrt{n}z_\alpha$. Den vanliga Students t-fördelning fås som ett specialfall av den icke-centriska med parametern 0. Percentiler för den icke-centriska t-fördelningen kan hämtas i Matlab med anropet `ntinv`.

(5.8) ger tillsammans med (5.7) ett nedåt begränsat konfidensintervall för κ_α med konfidensgrad β

$$\kappa_\alpha \geq \bar{x} - t'_{(n-1),\beta}(-\sqrt{n}z_\alpha) \frac{s}{\sqrt{n}} \quad (5.9)$$

En potentiell svaghet med Lawless intervallskattningsmetod är att ML-skattningen i (5.4) inte är väntevärdesriktig. Metoden uppges alltså fungera väl. I en artikel i *The American Statistician* [10] från 2007 jämförs Lawless metod med fem andra sätt att bilda konfidensintervall för normalpercentiler. Det konstateras att Lawless metod presterar i stort sett identiskt med en metod som bygger på en väntevärdesriktig ML-skattning. Lawless metod uppges också fungera väl för små och stora värden på α , det vill säga för α nära 0 eller 1.

Exempel 5.2.1 Givet är ett stickprov x_1, \dots, x_{18} där varje observerat värde antas komma från en normalfördelning med okänt väntevärde och okänd standardavvikelse. 5-percentilen $z_{0.05}$ i en $N(0, 1)$ standardfördelning är -1.6449 . Med 75 % konfidensgrad är stickprovets 5-percentil

$$x_{0.05} = \bar{x} - 8.2581 \frac{s}{\sqrt{18}}$$

där

\bar{x}	stickprovsmedelvärdet
8.2581	icke-centriska t-kvantilen $t'_{17, 0.75}(-\sqrt{18} \cdot (-1.6449))$
s	medelfelet $s = \sqrt{\frac{1}{17} \sum_1^{18} (x_i - \bar{x})^2}$

5.2.2 Simultana konfidensintervall för tre normala punktskattningar

Antag att man har tre stickprov $i = 1, 2, 3$ av storlekarna n_i från normala fördelningar $N(\mu_i, \sigma)$ med olika väntevärden och samma standardavvikelse. Låt s_i beteckna medelfelen för varje stickprov. Hela materialet kan då utnyttjas i skattningen av σ genom en sammanvägning eller så kallad pooling s_p [37]

$$s_p = \frac{(n_1 - 1)s_1 + (n_2 - 1)s_2 + (n_3 - 1)s_3}{(n_1 - 1) + (n_2 - 1) + (n_3 - 1)}$$

Ställ upp hypotesen

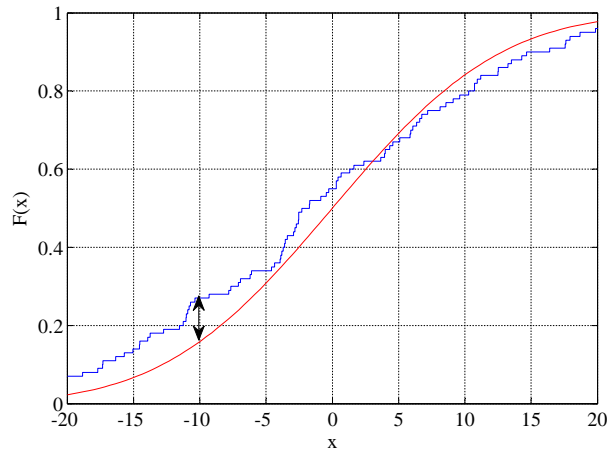
$$H_A : \mu_1 = \mu_2 = \mu_3$$

För att testa hypotesen på signifikansnivån α studeras konfidensintervallen för varje parvisa skillnad $\mu_i - \mu_j$, $i \neq j$. Om man med $f = n_1 + n_2 + n_3$ utgår från konfidensintervallen

$$I_{\mu_i - \mu_j} = \bar{x}_i - \bar{x}_j \pm t_{\alpha/2}(f) \cdot d$$

där $d = s_p \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}$ så är sannolikheten att alla $\binom{3}{2}$ kombinationer av konfidensintervall simultant täcker över $\mu_i - \mu_j$ inte lika med den önskade konfidensgraden $1 - \alpha$ utan mindre [7]. Rätt kombinerad konfidensgrad fås med Tukeys metod

$$I'_{\mu_i - \mu_j} = \bar{x}_i - \bar{x}_j \pm T_{\alpha/2}(3, f) \cdot s / \sqrt{n}$$



Figur 5.3: Illustrationsexempel av empirisk och teoretisk fördelningsfunktion där teststatistikan i ett Kolmogorov-Smirnov-test markerats.

där $f = 3(n - 1)$ och T_α är α -kvantilen i en fördelning som på engelska kallas *studentized range distribution*. Om stickproven är olika stora men inte alltför olika fungerar Tukeys metod i regel bra med n ersatt av något medelvärde av n_i :na [7]. Tukeys metod används per standard i Matlab-rutinen `multcompare` som beräknar simultana konfidensintervall för normala punktskattningar.

5.2.3 Kolmogorov-Smirnovs test

För att testa om data kommer från en viss fördelning kan Kolmogorov-Smirnovs test användas. Det är en icke-parametrisk metod, ett så kallat goodness-of-fit test. Låt $F_e(x)$ beteckna den empiriska fördelningsfunktionen och $F_0(x)$ den teoretiska. Hypotesen är

$$H_0 : F_e(x) = F_0(x)$$

$$H_1 : F_e(x) \neq F_0(x)$$

Idén är att beräkna den största absolutavvikelsen mellan den empiriska fördelningsfunktionen och den teoretiska

$$T_{KS} = \sup_x |F_e(x) - F_0(x)|$$

Om T_{KS} är större än någon kritisk gräns förkastas H_0 . Teststatistikan illustreras i Figur 5.3. [30]

6 Resultat

6.1 Verifiering och kalibrering genom provning och simulering

6.1.1 Provningsresultat

Totalt dragbelastades 21 provkroppar. Provkropparnas geometri beskrivs i Tabell 3.2. Förloppet tillsammans med brottmod illustreras med skisser i Figur 6.1. Provkropparna är indelade i fyra grupper A)-D) efter hur de gick till brott. A) I tre fall av 21 observerades ingen spricka vid urtagen under själva dragprovningen och dessa provkroppar stryks från analysen. I efterhand upptäcktes ändå mycket små sprickor vid urtagen i två av de tre fallen. I ett av de tre fallen sprack provkroppen mitt på längden i höjd med en kvist och inte vid något urtag.

Vid 18 av 21 provningar observerades uppsprickning. Brottmoden för de 18 provkropparna var B) i nio fall sprött uppsprickningsbrott vilket illustreras i Figur 6.2a. C) I två fall propagerade en lång spricka mer eller mindre parallellt med längsriktningen och dragbrott skedde framför urtaget. En bild på dessa provkroppar visas i Figur 6.2b. I två andra fall observerades en långsamt propagerande spricka men dragbrott nära urtaget inträffade innan sprickan ledde till brott. För en av dessa två kroppar uppstod sprickor vid båda urtagen så pass kort tid efter varandra att det ej var möjligt att avgöra vilken spricka som kom först. D) I resterande fem av de 18 fallen observerades ingen brottmod då belastningen avbröts efter en utdragen och seg uppsprickning. Även i ett av dessa fem fall uppstod sprickor vid båda urtagen samtidigt. Uppsprickningsförloppet var segt och uppsprickningslasten hög.

I Tabell 6.1 finns uppmätt brottlast, brottspänning och en kommentar om brottmoden för varje provkropp. För samtliga provkroppar återges tid-kraft-diagram i Bilaga 1. Förutom att provkropparna dragprovades så uppmättes även densiteten. Medelvärde uppgick till 392.3 kg/m^3 och variationskoefficienten var 8.2 %.

6.1.2 Verifiering av beräkningsformel för två likadana urtag med centrisk belastning

Fall 1 (en spricka)

Spricklasten med två likadana urtag och centrisk last ges av (4.5) 52 som med insättning av $\beta = 0.5$, $e_P = 0$ och $\widetilde{K}_c = \sqrt{G_c E_L}$ blir

$$P_c = \widetilde{K}_c \frac{b\sqrt{2h}}{\sqrt{-1 + \frac{2}{1+\alpha} + 6\frac{(1-\alpha)^2}{(1+\alpha)^3}}}$$

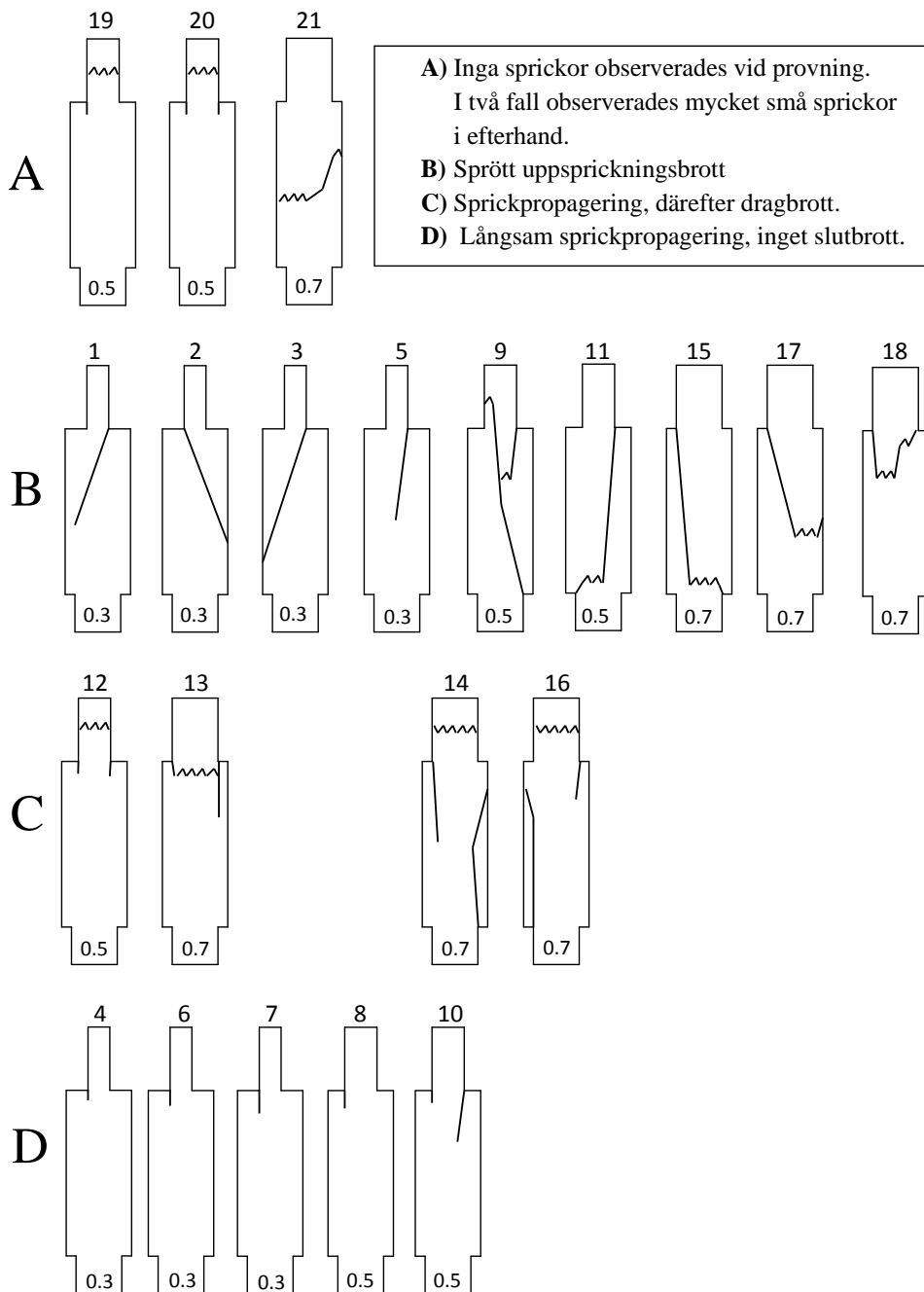
Brottlasterna från provningen används för att skatta provkropparnas skenbara brottseghet \widetilde{K}_c . I Tabell 6.2 redovisas skattade värden på \widetilde{K}_c för de 18 provningar då uppsprickning vid urtag förekom. Utifrån dessa värden bestäms medelvärde samt karakteristiskt värde på \widetilde{K}_c , vilka uppgår till 2.11 MN/m^{3/2} respektive 1.50 MN/m^{3/2}. En graf med residualer visas i Figur 6.4a, och i Figur 6.4b visas en lognormalfördelningsplot. Med figuren troliggörs antagandet om lognormalfördelad data. Ett Kolmogorov-Smirnov-test är inte signifikant på 5 %-nivån med hypotesen att $\ln \widetilde{K}_{c,i}$ för $i = 1, \dots, 18$ är observationer på en normalfördelning. Därmed är hypotesen inte heller styrkt. Men det förefaller inte finnas någon invändning mot antagandet om lognormalfördelad data. I Figur 6.5 jämförs uppmätt spricklast med teoretisk, då beräknat medelvärde på den skenbara brottsegheten $\widetilde{K}_{c,m}$ används. En linjär regressionsanalys ger förhållandet 0.99 mellan uppmätt spricklast och teoretisk. Förklaringsgraden R^2 är 70 %.

Fall 2 (två sprickor)

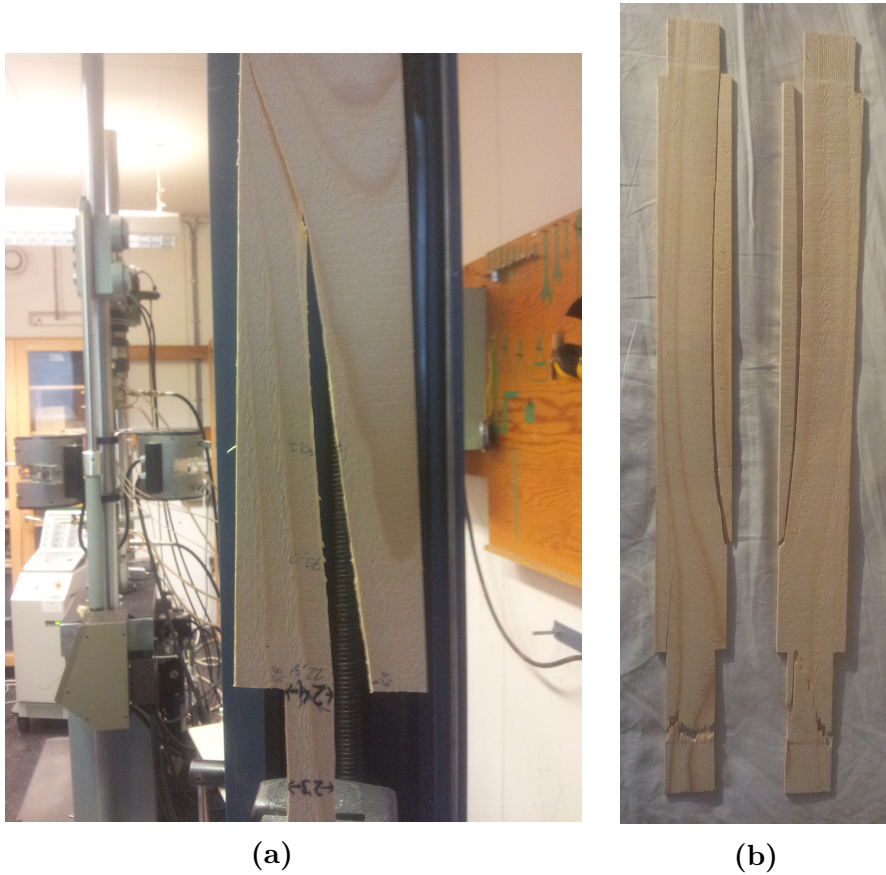
Spricklasten med två likadana urtag och centrisk last ges av (4.8) sidan 52 som med insättning av $\widetilde{K}_c = \sqrt{G_c E_L}$ (jämför Definition 5.1) blir

$$P_c = \widetilde{K}_c \frac{b\sqrt{4h}}{\sqrt{(1-\alpha)/\alpha}}$$

Brottlasterna från de 18 provningar då uppsprickning vid urtag förekom används för att skatta den skenbara brottsegheten \widetilde{K}_c och resultatet finns i Tabell 6.3. Medelvärdet på den skenbara brottsegheten $\widetilde{K}_{c,m}$ uppgår till 1.72 MN/m^{3/2} och det karakteristiska värdet uppgår till 1.19 MN/m^{3/2}. I Figur 6.3 visas brottlasterna från provningen tillsammans med beräknade brottlaster enligt Fall 1 och Fall 2. För beräkningarna har



Figur 6.1: Skisserade brottmoder. Provkropparna är numrerade i enlighet med Tabell 3.2. Sågtandade linjer markerar dragbrott och provkropparnas ungefärliga α -värden anges längst ner på varje provkropp i figuren.



Figur 6.2: a) Provkropp nr 3 som gått till brott genom uppsprickning. b) Provkroppar nr 14 och 16 för vilka uppsprickning ej har gett upphov till brott. Istället har dragbrott inträffat. Notera att sprickorna ej korsar träets fiberriktning. Uppsprickning i dessa två fall skedde vid den ände som ej var tänkt för uppsprickning.

medelvärden på b , h samt respektive medelvärde på \widetilde{K}_c använts. En graf med residualer visas i Figur 6.6a och i Figur 6.6b finns en lognormalfördelningsplot. Med figuren troliggörs antagandet om lognormalfördelad data. Ett Kolmogorov-Smirnov-test är inte signifikant på 5 %-nivån med hypotesen att $\ln \widetilde{K}_{c,i}$ för $i = 1, \dots, 18$ är observationer på en normalfördelning. Därmed är hypotesen inte heller styrkt. Men det förefaller inte finnas någon invändning mot antagandet om lognormalfördelad data. I Figur 6.7 jämförs uppmätt spricklast med teoretisk, då beräknat medelvärde på den skenbara brottsegheten $\widetilde{K}_{c,m}$ används. En linjär regressionsanalys ger förhållandet 1.01 mellan uppmätt spricklast och teoretisk. Förklaringsgraden R^2 är 73 %.

6.1.3 Kalibrering av numeriska beräkningsmodeller genom provning

Noduppläppning

Kvoten mellan uppmätt och simulerad brottlast som med slumpmodellen (4.12) benämns D , redovisas i Tabell 6.4. Materialdatan för simuleringen finns i Tabell 4.1 och i första raden i Tabell 2.2. Beräkningskoefficienten D^* uppgår till 0.850 och det karakteristiska värdet uppgår till 0.581. I Figur 6.8a visas residualerna för D^* och i Figur 6.8b visas en lognormalfördelningsplot av de 18 skattningarna av D . Residualerna visar ingen trend eller tydligt mönster. Då simulerade brottlaster korrigeras med beräkningskoefficienten D^* förhåller de sig till uppmätta värden enligt Figur 6.9 och en linjär regressionsanalys ger sambandet 0.975 med förklaringsgraden 70 %. Med lognormalfördelningsplotten troliggörs antagandet om lognormalfördelad data. Ett Kolmogorov-Smirnov-test är inte signifikant på 5 %-nivån med hypotesen att $\ln D_i$ för $i = 1, \dots, 18$ är observationer på en normalfördelning. Därmed är hypotesen inte heller styrkt. Men det förefaller inte finnas någon invändning mot antagandet om lognormalfördelad data.

Kohesiva zonmodellen

Kvoten mellan uppmätt och simulerad brottlast som med slumpmodellen (4.12) benämns D , redovisas i Tabell 6.5. Materialdatan för simuleringen finns i Tabell 4.2 och i första raden i Tabell 2.2. Beräkningskoefficienten D^* uppgår till 0.854 och det karakteristiska värdet är 0.590. I

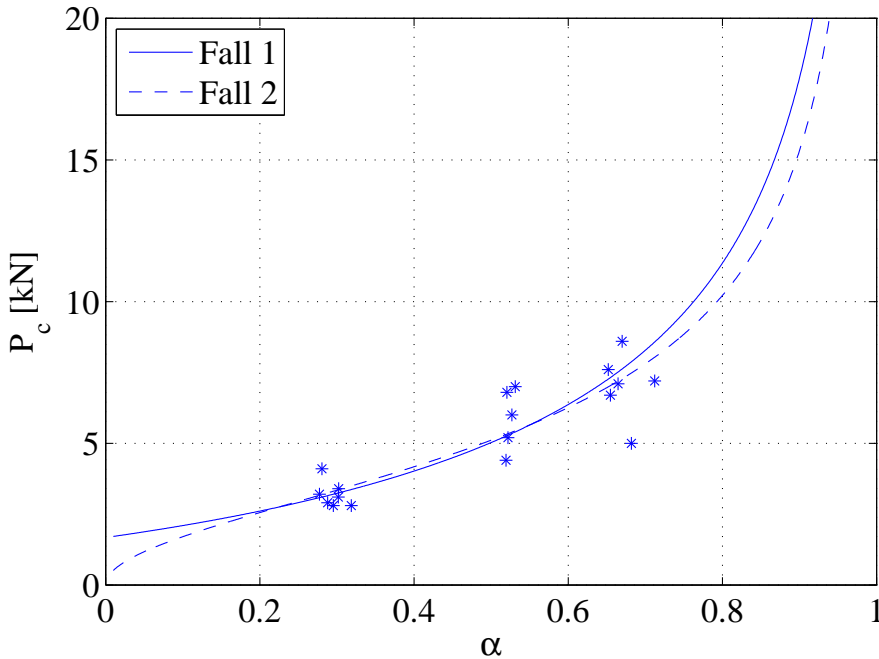
Figur 6.10a visas residualerna för D^* och i Figur 6.10b visas en lognormalfördelningsplot för de 18 skattningarna av D . Residualerna påvisar ingen trend eller tydligt mönster. Då simulerade brottlaster korrigeras med beräkningskoefficienten D^* förhåller de sig till uppmätta värden på brottlast enligt Figur 6.11. Jämfört med en linjär regressionsanalys fås sambandet 0.987 och förklaringsgraden är 71 %. Med lognormalfördelningsplotten troliggörs antagandet om lognormalfördelad data. Ett Kolmogorov-Smirnov-test är inte signifikant på 5 %-nivån med hypotesen att $\ln D_i$ för $i = 1, \dots, 18$ är observationer på en normalfördelning. Därmed är hypotesen inte heller styrkt. Men det förefaller inte finnas någon invändning mot antagandet om lognormalfördelad data.

6.1.4 Bestämning av den skenbara brottsegheten \widetilde{K}_c genom simulering

Vid simuleringar med noduppsläppningsmetod erhålls värden på andelen Mod I, se (4.9), enligt Figur 6.12 då α , β och e_P/h varieras. Andelen Mod I antas konstant lika med 0.6 då $\alpha \geq 0.8$. Antagandet är rimligt med tanke på resultatet i Figur 6.12. Om α hålls konstant kan andelen Mod I:s variation med β och e_P illustreras med ytor i rummet. I Figur 6.13, Figur 6.14 och Figur 6.15 visas sådana ytor för $\alpha = 0.5$, $\alpha=0.6$ respektive $\alpha = 0.7$. Ytornas ekvationer ges i figurtexterna. Korrelationskoefficienten är över 97 %.

6.1.5 Modblandningen i några speciella fall

I Figur 6.16 illustreras andelen Mod I för fallet då $\beta = 0.5$ och $e_P = 0$ det vill säga då urtagen är två stycken (likadana) och belastningen är centrisk. Kurvan visar vad avsnitt 6.1.4 ger för andel Mod I då α varieras, och punkterna visar simuleringens resultat. Effekten av en excentricitet ($e_P/h = 0.0125$) illustreras i Figur 6.17. För fallet då $\beta = 0$ och $e_P = 0$ dvs ensidigt urtag med centrisk belastning beror andelen Mod I på α enligt Figur 6.18. Effekten av en excentricitet ($e_P/h = 0.0125$) visas i Figur 6.19.



Figur 6.3: Uppmätta brottlaster visas som punkter i diagrammet. De härledda beräkningsuttryckens α -beroende visas med heldragen linje för Fall 1 och med streckad linje för Fall 2. För de båda beräkningsformlerna har medelvärden på bredd och tvärsnittshöjd ($b = 5.63$ mm, $h = 69.5$ mm) använts, samt beräknade medelvärden på den skenbara brottsegheten $\widetilde{K}_{c,m} = 2.11$ MN/m^{3/2} för Fall 1 respektive $\widetilde{K}_{c,m} = 1.72$ MN/m^{3/2} för Fall 2.

6.1.6 Verifiering av beräkningsformel allmänna fallet

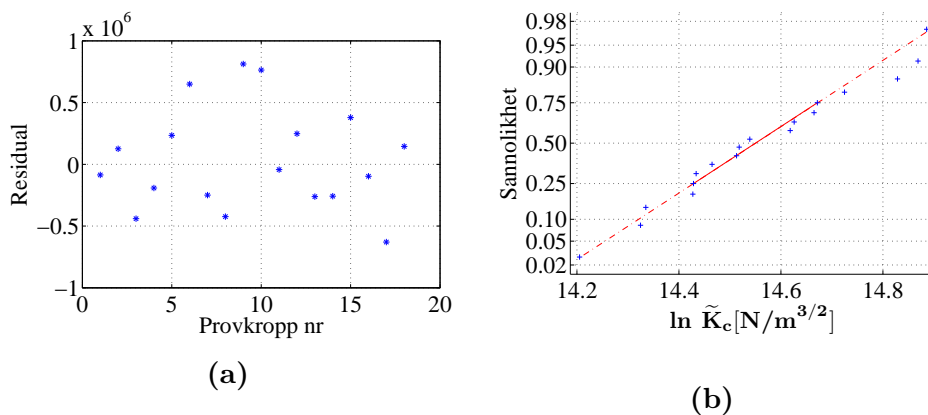
Handberäkningsformeln (4.5) sidan 52 verifieras genom simulering med den numeriska beräkningsmodellen (noduppsläppning) som kalibrerats med det karakteristiska värdet på beräkningskoefficienten $D^* = 0.5811$. Sambandet mellan simulering och beräkningsuttryck beskrivs med hjälp av ett medelvärde 0.56. Alltså ska handberäkningsformeln för det allmänna fallet multipliceras med faktorn 0.56 för att ge det karakteristiska värdet för brottsegheten. Vid verifieringen antas andelen Mod I vara lika med 0.6 för alla värden på α , β och e_P/h .

Tabell 6.1: Brottlast, brottspänning och kommentar om brottmod för de 21 provkropparna. Brottmoden beskrivs vidare i Figur 6.1. Brottspänningen har beräknats framför urtaget. Medelvärden och variationskoefficienter anges för varje provgrupp.

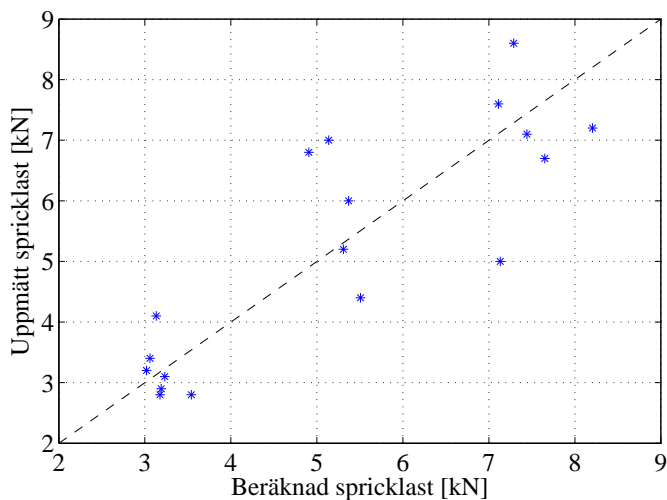
	Prov nr	Brottlast P [kN]	Brottspänning f_t [MPa]	Brottmod
Grupp 1 $\alpha \approx 0.30$	1	3.1	26	B
	2	3.2	30	B
	3	2.8	21	B
	4	2.9	25	D
	5	3.4	30	B
	6	4.1	37	D
	7	2.8	24	D
<i>Medelvärde</i>		<i>3.2</i>	<i>28</i>	
<i>Variationskoeff.</i>		<i>0.14</i>		
Grupp 2 $\alpha \approx 0.50$	8	4.4	21	D
	9	6.8	36	B
	10	7.0	36	D
	11	5.2	25	B
	12	6.0	29	C
	19	6.5	28	A
	20	6.4	30	A
<i>Medelvärde</i>		<i>6.0</i>	<i>29</i>	
<i>Variationskoeff.</i>		<i>0.15</i>		
Grupp 3 $\alpha \approx 0.70$	13	6.7	25	C
	14	7.2	27	C
	15	8.6	34	B
	16	7.1	28	C
	17	5.0	21	B
	18	7.6	30	B
	21	7.8	28	A
<i>Medelvärde</i>		<i>7.1</i>	<i>28</i>	
<i>Variationskoeff.</i>		<i>0.16</i>		

Tabell 6.2: Skattad skenbar brottseghet för aktuella provkroppar när uppmätta brottlaster jämförs med teoretiska enligt handberäkningsformel Fall 1 (en spricka).

	Provkroppsnr	$\ln \widetilde{K}_c[\text{N/m}^{\frac{3}{2}}]$	$\widetilde{K}_c[\text{N/m}^{\frac{3}{2}}]$
Grupp 1	1	14.52	
	2	14.62	
	3	14.33	
	4	14.46	
	5	14.67	
	6	14.83	
	7	14.43	
Grupp 2	8	14.34	
	9	14.89	
	10	14.87	
	11	14.54	
	12	14.67	
Grupp 3	13	14.63	
	14	14.43	
	15	14.43	
	16	14.73	
	17	14.51	
	18	14.21	
Medelvärde		14.56	$2.106 \cdot 10^6$
Variationskoeff.		0.01321	$0.1863 \cdot 10^6$
Karakteristiskt värde		14.19	$1.449 \cdot 10^6$



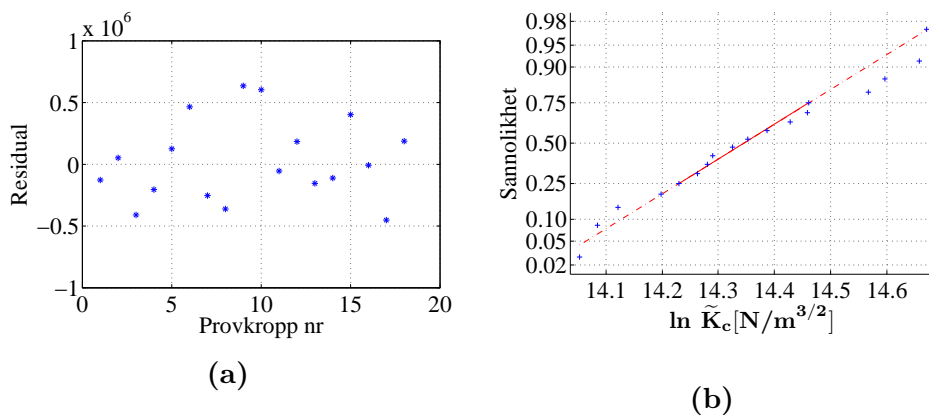
Figur 6.4: a) Avvikelser från medelvärdet av skattade skenbara brottsegheter då uppmätt brottlast jämförs med teoretisk enligt handberäkningsformel Fall 1 (en spricka). b) Lognormalfördelningsplot för skattade värden på \tilde{K}_c när handberäkningsformel Fall 1 används.



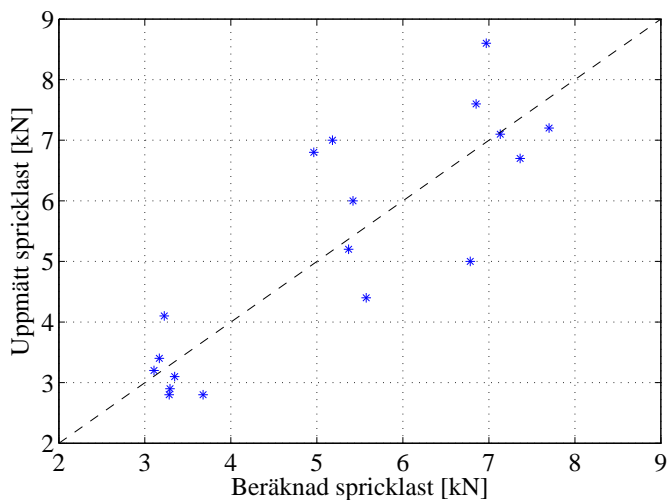
Figur 6.5: Uppmätta och teoretiska värden (handberäkningsformel Fall 1) på spricklast då beräknat medelvärde på den skenbara brottsegheten $\tilde{K}_{c,m} = 2.11 \text{ MN/m}^{3/2}$ används. Den streckade linjen har lutningen 1.

Tabell 6.3: Skattad skenbar brottseghet för aktuella provkroppar när uppmätta brottlaster jämförs med teoretiska enligt handberäkningsformel Fall 2 (två sprickor).

	Provkroppsnr	$\ln \widetilde{K}_c[\text{N/m}^{\frac{3}{2}}]$	$\widetilde{K}_c[\text{N/m}^{\frac{3}{2}}]$
Grupp 1	1	14.28	
	2	14.39	
	3	14.08	
	4	14.23	
	5	14.43	
	6	14.60	
	7	14.20	
Grupp 2	8	14.12	
	9	14.67	
	10	14.66	
	11	14.33	
	12	14.46	
Grupp 3	13	14.26	
	14	14.29	
	15	14.57	
	16	14.35	
	17	14.05	
	18	14.46	
Medelvärde		14.36	$1.719 \cdot 10^6$
Variationskoeff.		0.01312	$0.1889 \cdot 10^6$
Karakteristiskt värde		13.99	$1.191 \cdot 10^6$



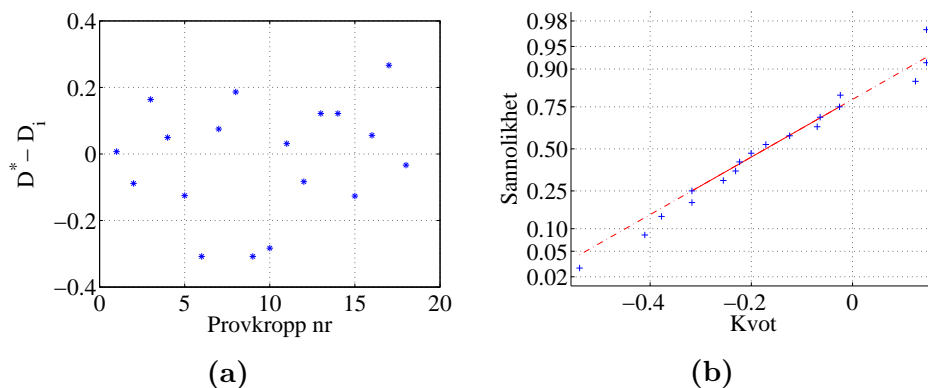
Figur 6.6: a) Avvikelser från medelvärdet av skattade skenbara brottsegheter då uppmätt brottlast jämförs med teoretisk enligt handberäkningsformel Fall 2 (två sprickor). b) Lognormalfördelningsplot för skattade värden på \tilde{K}_c när handberäkningsformel Fall 2 används.



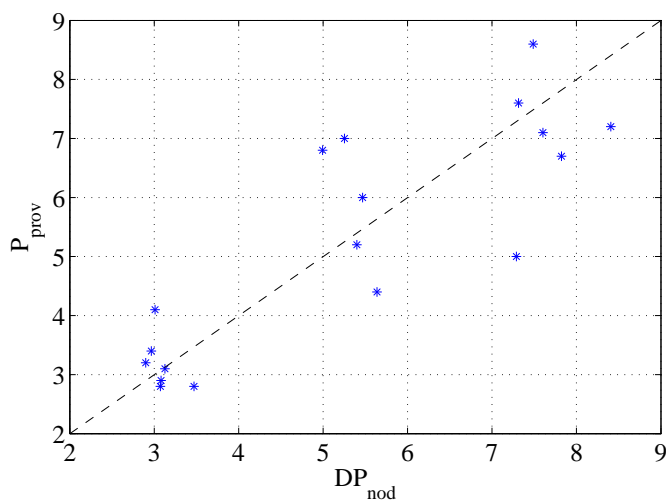
Figur 6.7: Uppmätta och teoretiska värden (handberäkningsformel Fall 2) på spricklast då beräknat medelvärde på den skenbara brottsegheten $\tilde{K}_{c,m} = 1.72 \text{ MN/m}^{3/2}$ används. Den streckade linjen har lutningen 1.

Tabell 6.4: Kvot mellan simulerad och uppmätt brottlast för aktuella provkroppar när noduppsläppningsmetod används.

	Provkroppsnr	$(\ln D)_{noduppsl}$	$D_{noduppsl}$
Grupp 1	1	-0.1712	
	2	-0.06392	
	3	-0.3775	
	4	-0.2231	
	5	-0.02565	
	6	0.1463	
	7	-0.2554	
Grupp 2	8	-0.4105	
	9	0.1463	
	10	0.1246	
	11	-0.2002	
	12	-0.06963	
Grupp 3	13	-0.1242	
	14	-0.3174	
	15	-0.3174	
	16	-0.02407	
	17	-0.2310	
	18	-0.5394	
Medelvärde		-0.1630	0.8496
Variationskoeff.		-1.197	0.1915
Karakteristiskt värde		-0.5428	0.5811



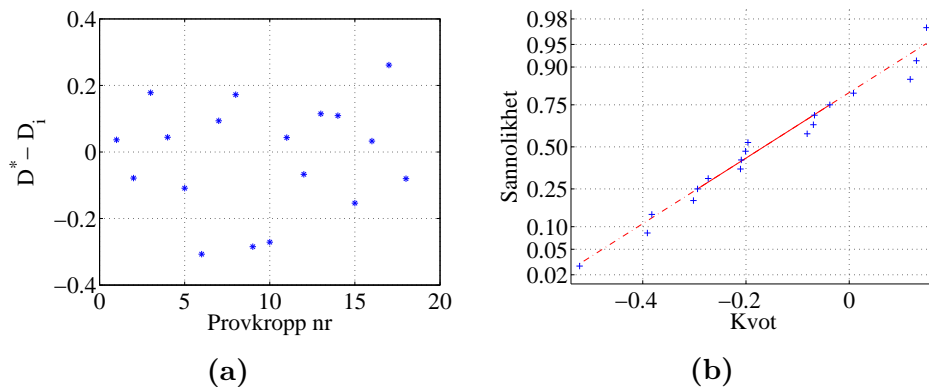
Figur 6.8: a) Avvikelser från medelvärdet av kvoten $D_i = P_{prov,i}/P_{noduppsl,i}$ då uppmätt brottlast jämförs med simulerad brottlast enligt noduppsläppningsmetod. b) Lognormalfördelningsplot av kvoten mellan uppmätt och simulerad brottlast när noduppsläppningsmetod används.



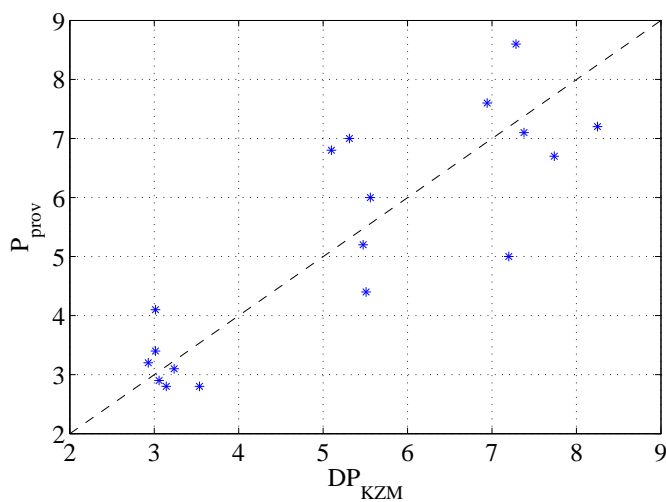
Figur 6.9: Uppmätta och simulerade brottlaster då beräkningskoefficienten $D = 0,850$ och noduppsläppning används. Den streckade linjen har lutningen 1.

Tabell 6.5: Kvot mellan simulerad och uppmätt brottlast för aktuella provkroppar när koheziva zonmodellen används.

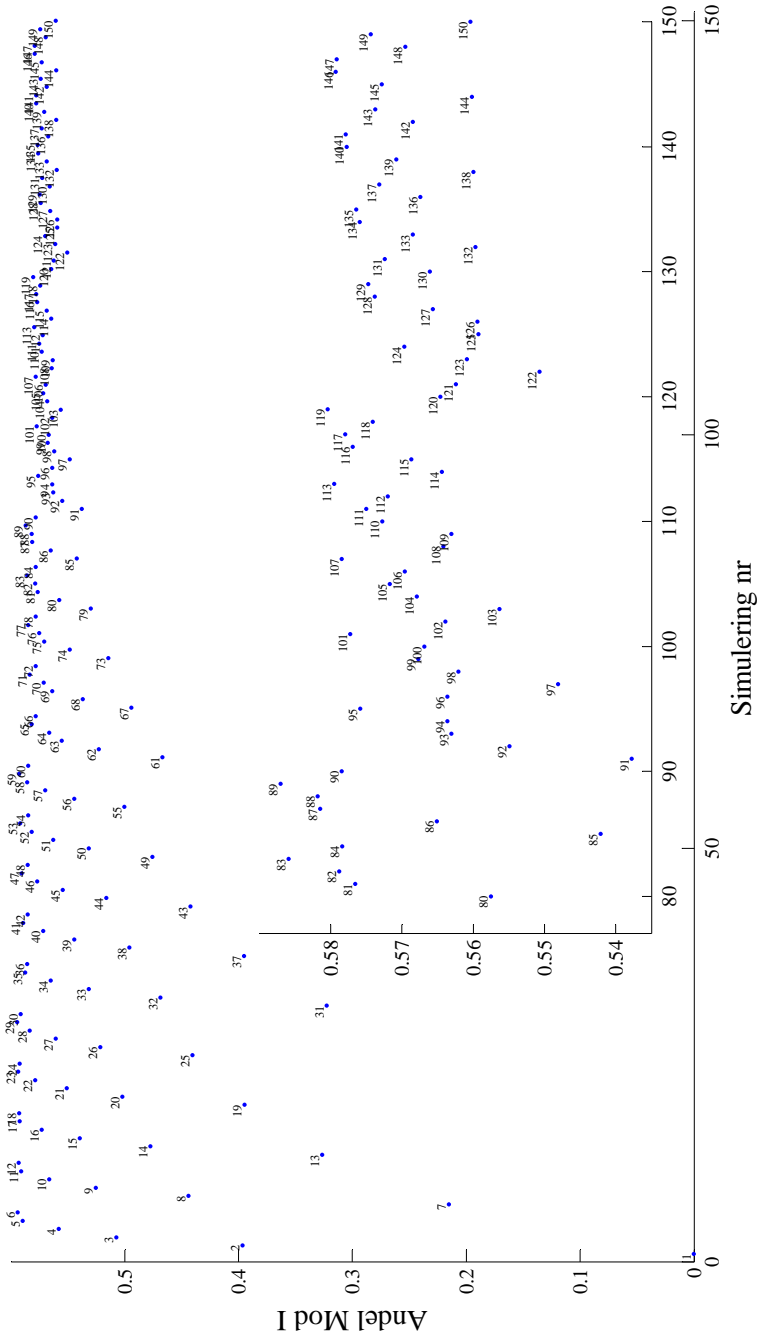
Provkroppsgrupp	Provkroppsnummer	$(\ln D)_{KZM}$	D_{KZM}
Grupp 1	1	-0.06741	
	2	-0.2010	
	3	-0.06941	
	4	-0.3911	
	5	-0.2107	
	6	-0.03752	
	7	0.1497	
Grupp 2	8	-0.2733	
	9	-0.3825	
	10	0.1302	
	11	0.1181	
	12	-0.2092	
Grupp 3	13	-0.08158	
	14	-0.3018	
	15	-0.2939	
	16	0.008173	
	17	-0.1963	
	18	-0.5224	
Medelvärde		-0.1573	0.8544
Variationskoeff.		-1.211	0.1871
Karakteristiskt värde		-0.5280	0.5898



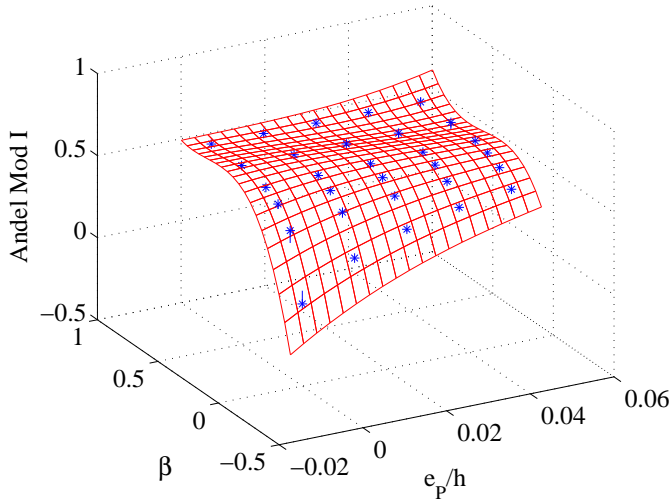
Figur 6.10: a) Avvikelser från medelvärdet av kvoten $D_i = P_{prov,i}/P_{KZM,i}$ då uppmätt brottlast jämförs med simulerad brottlast enligt kohesiv zonmetod. b) Lognormalfördelningsplot av kvoten mellan uppmätt och simulerad brottlast när kohesiv zonmetod används.



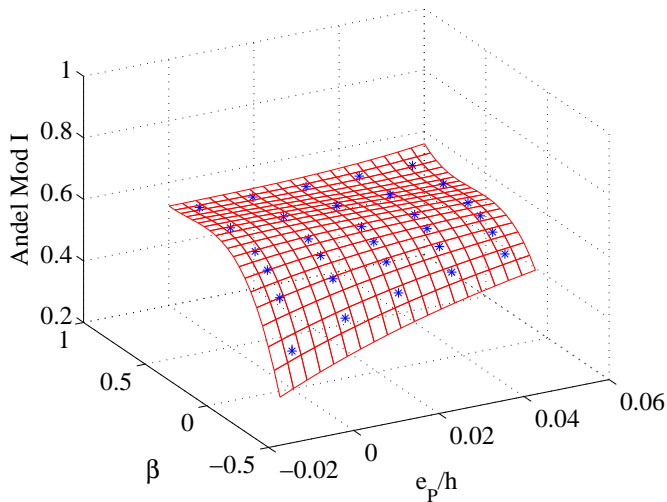
Figur 6.11: Uppmätta och simulerade brottlaster då beräkningskoefficienten $D = 0.854$ och kohesiv zonmodell används. Den streckade linjen har lutningen 1.



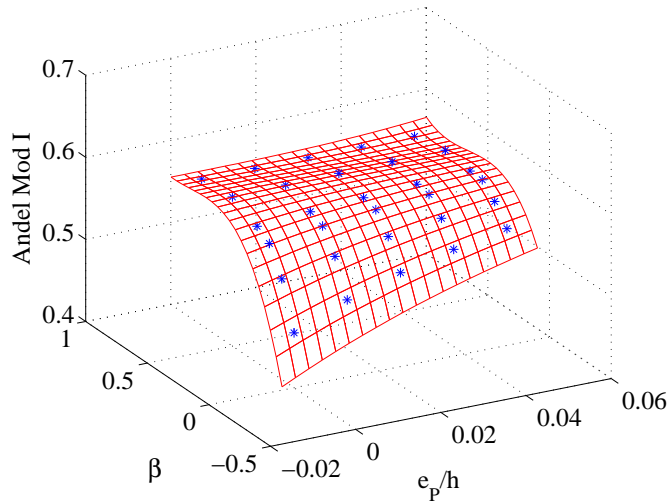
Figur 6.12: Andel Mod I vid uppsprickning då α , β och e_P/h varieras. Totalt har 150 simuleringar gjorts, där värdena på β varieras cykliskt mellan 0, 0.1, 0.2, 0.3, 0.5 och 0.75. e_P/h varieras cykliskt var sjätte simulering mellan 0, 0.0125, 0.025, 0.0375, 0.05 och α varieras vart 30:e simulering mellan 0.5, 0.6, 0.7, 0.8 och 0.9. Till exempel gäller för simulering nr 7 att $\beta=0$, $e_P/h=0.0125$ och $\alpha=0.5$, och för simulering nr 50 att $\beta=0.1$, $e_P/h=0.0375$ och $\alpha=0.6$



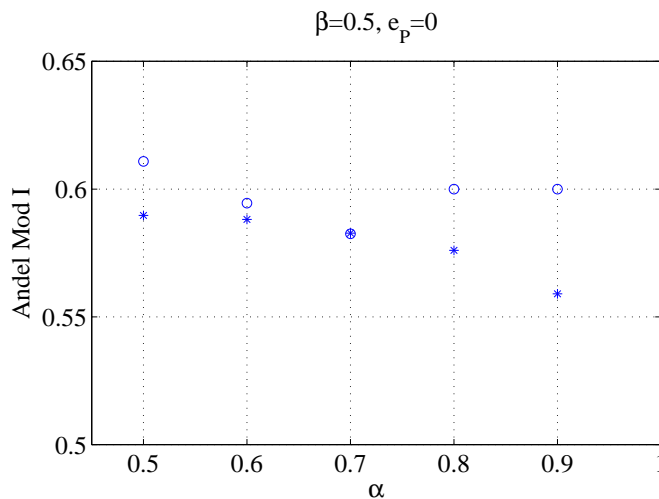
Figur 6.13: Andel Mod I som funktion av β och e_P/h då $\alpha = 0.5$. Ytan som är anpassad efter mätpunkterna har ekvationen $\text{Mod I} = 0.08048 + 12.16e_P/h + 2.882\beta - 134(e_P/h)^2 - 39.9e_P/h\beta - 5.06\beta^2 + 652(e_P/h)^3 + 159.2(e_P/h)^2\beta + 30.95e_P/h\beta^2 + 2.835\beta^3$. Residualavstånden markeras med heldragen linje.



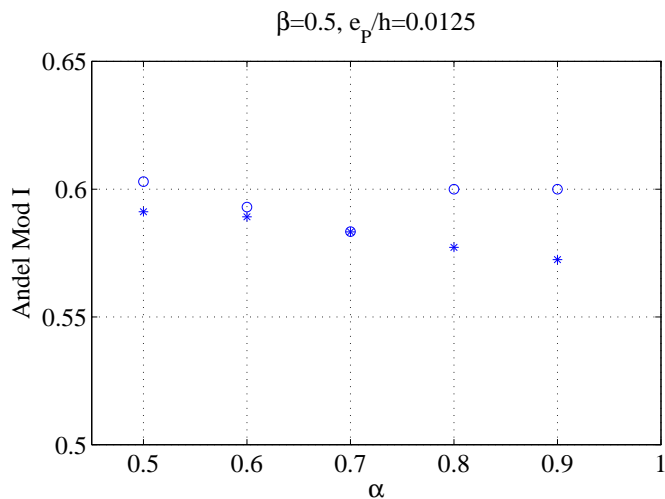
Figur 6.14: Andel Mod I som funktion av β och e_P/h då $\alpha = 0.6$. Ytan som är anpassad efter mätpunkterna har ekvationen $\text{Mod I} = 0.3435 + 4.679e_P/h + 1.259\beta - 39.62(e_P/h)^2 - 15e_P/h\beta - 2.041\beta^2 + 155.3(e_P/h)^3 + 50.49(e_P/h)^2\beta + 11.43e_P/h\beta^2 + 1.054\beta^3$. Residualavstånden markeras med heldragen linje.



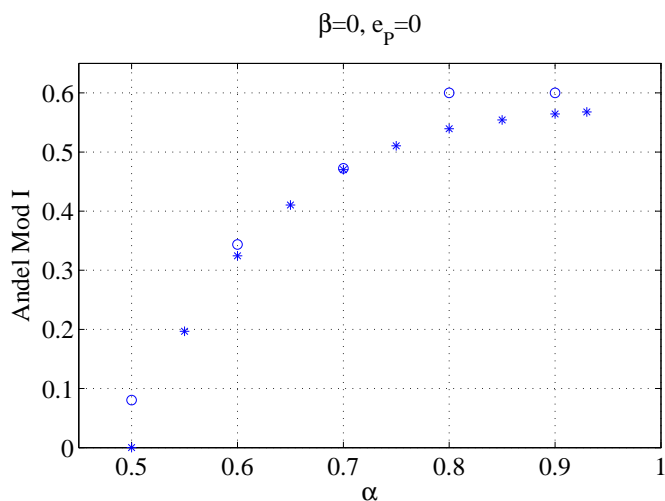
Figur 6.15: Andel Mod I som funktion av β och e_P/h då $\alpha = 0.7$. Ytan som är anpassad efter mätpunkterna har ekvationen $\text{Mod I} = 0.4724 + 1.96e_P/h + 0.5457\beta - 13.63(e_P/h)^2 - 5.676e_P/h\beta - 0.8742\beta^2 + 44.88(e_P/h)^3 + 17.53(e_P/h)^2\beta + 4.02e_P/h\beta^2 + 0.4463\beta^3$. Residualavstånden markeras med heldragen linje.



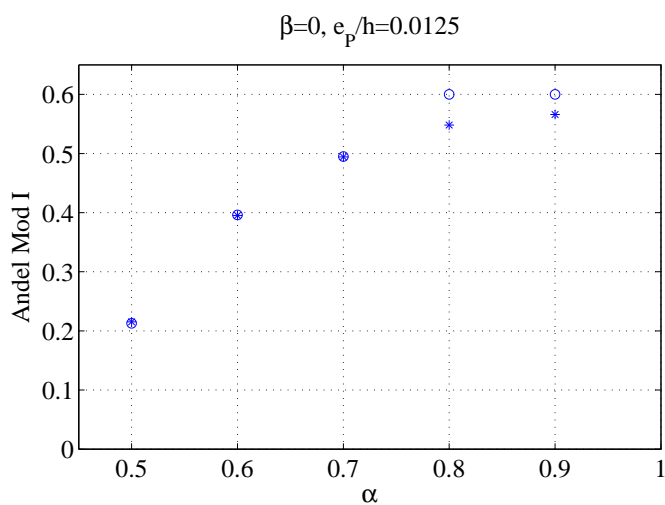
Figur 6.16: Andel Mod I vid uppsprickning då $\beta = 0.5$ och $e_P = 0$. Stjärnor visar simuleringsresultat och cirklar beräkningsresultat enligt avsnitt 6.1.4.



Figur 6.17: Andel Mod I vid uppsprickning då $\beta = 0.5$ och $e_P/h = 0.0125$. Stjärnor visar simuleringsresultat och cirklar beräkningsresultat enligt avsnitt 6.1.4.



Figur 6.18: Andel Mod I vid uppsprickning då $\beta = 0$ och $e_P = 0$. Stjärnor visar simuleringsresultat och cirklar beräkningsresultat enligt avsnitt 6.1.4.



Figur 6.19: Andel Mod I vid uppsprickning då $\beta = 0$ och $e_P/h = 0.0125$. Stjärnor visar simuleringsresultat och cirklar beräkningsresultat enligt avsnitt 6.1.4.

7 Slutsatser

7.1 Diskussion

Provningsen visar att det går att förutsäga uppsprickningslasten med ett praktiskt beräkningsuttryck när provkropparna är små, har två likadana urtag och lasten är centrisk. Vid provning observerades att de propagerande sprickorna i regel inte korsar trämateriallets fibrer. Detta kan ge en uppfattning om en eventuell sprickas riktning innan sprickan har uppstått. En spricka som propagerar in mot tvärsnittets mitt leder till att tråkroppen kollapsar. En spricka som propagerar mot tvärsnittets kant leder inte direkt till brott. Provningsen visade också att brottmoden är mycket känslig för förekomsten av kvistar. För en verklig konstruktionsdel med urtag kommer kvistar och andra svagheter både i närheten av urtagszonen och en bit därifrån att göra det svårt att avgöra vilken sorts brott som först inträffar. Att dra generella slutsatser från resultat på felfria provkroppar innebär en praktisk utmaning. Larsen & Riberholt [29] uppger att kvistar ökar brottsegheten lokalt. Vid provning av felfria tråkroppar bör i så fall förhållandet mellan uppsprickningsrisk och dragbrottsrisk vara maximalt. Slutsatserna från provningsen bör därmed vara på säkra sidan. Troliga felkällor vid provningsen är:

- Infästningen var kanske inte perfekt ledad
- Provkropparna var inte perfekt symmetriskt utformade kring sin längsaxel

Dragningen kan ha gett upphov åt ett litet moment på grund av ovanstående orsaker. Frågan är hur stort moment som kan ha uppstått och i så fall hur mycket det inverkar på resultaten. Det bör ge resultat på säkra sidan eftersom momentet lär sänka hållfastheten med avseende på uppsprickning.

I härledningen av beräkningsuttrycket är det möjligt att anta förekomsten av en eller två simultana sprickor om urtagen är två stycken. Provningsen visar emellertid att det är osannolikt med två simultana sprickor vid urtagen. Det är alltså i överensstämmelse med empirin att använda beräkningsuttrycket med antagande om en spricka. Dessutom ger antagandet om en spricka ett värde på \tilde{K}_c som ligger närmare det värde som fås enligt litteraturen än vad antagandet om två simultana sprickor gör. Beräkningsuttrycket med en spricka är alltså att föredra. Med E , $G_{c,I}$ och $G_{c,II}$ enligt [3] blir $\tilde{K}_c = 2.45 \text{MN/m}^{\frac{3}{2}}$ om andelen Mod I antas vara 60 % och G_c ges av (4.10). Detta är högre än de

skattade värdena 1.72 respektive 2.11 MN/m^{3/2}. I FE-simuleringarna var modblandningen ungefär densamma (andelen Mod I $\approx 60\%$) oavsett om en eller två simultana sprickor modellerades. Wernersson antog vid undersökningar av limfogar att för tvåskäriga förband med symmetrilinje enligt Figur 7.3 och med centrisk dragbelastning, fås uppsprickning i de båda limfogarna samtidigt [54]. Men jämförelse av de teoretiska ekvationerna som härletts i detta arbete för uppsprickning vid ett respektive två urtag visar att om $\alpha > 0.1$ blir spricklasten lägre om endast en spricka antas. Med tanke på detta hade Wernersson fått resultat som varit mer på säkra sidan om han hade räknat med uppsprickning på ena sidan enbart.

I det allmänna fallet med (excentrisk) belastning och ett eller två (olika djupa) urtag ges en god överensstämmelse mellan simuleringsresultaten och beräkningsuttrycket. Beräkningsuttrycket bygger på förenklingen att andelen Mod I är 60%. För små till medelstora urtag är detta en mycket god uppskattning men då $\alpha \lesssim 0.65$ riskerar andelen Mod I att överskattas om urtagen är olika stora eller om urtaget är ensidigt. Det ger i alla fall resultat på säkra sidan att ansätta andelen Mod I som 60%. Alternativet är att bestämma andelen med viss möda och insätta i uttrycket för \widetilde{K}_c . De teoretiska beräkningsuttrycken visar att bärförmågan med avseende på uppsprickning inte växer lika snabbt som funktion av tvärsnittshöjden som bärförmågan med avseende på dragbrott gör. Risken för brott på grund av uppsprickning förefaller alltså bli större vid en ökning av tvärsnittshöjden. För övrigt är storlekseffekten inte utredd eller verifierad genom provning. Det går alltså inte att urskillningslöst tillämpa det härledda beräkningsuttrycket på stora konstruktionsdelar.

För konstruktionsvirke kan fiberriktningen relativt enkelt observeras och sprickans tillväxtriiktning i viss mån förutsägas. För limträ där fiberriktningen varierar mellan lamellerna är det oklart hur tillväxtriiktningen kan bestämmas. Eftersom limfogen är starkare än materialet runtomkring så är det möjligt att en spricka som når limfogen med en inte alltför stor angreppsvinkel, tillväxer parallellt med limfogen.

Simulering med kohesiv zon ger i stort sett samma resultat som simulering med noduppsläppningsmetod. Slutsatsen är att en linjärelastisk analys räcker för att beskriva verkningssättet. Felkällor i de numeriska beräkningsmodellerna är i första hand indatan: värden på E-modul, brottenergi och i kohesiva zonmodellen hållfastheter. Det skattade sambandet mellan provning och simulering är 0.85 (medelvärde). Simuleringen överskattar alltså bärförmågan. Skillnaden kan bero på att

simulering med FEM generellt ger styvare respons och på att felkällorna vid provning verkar för att minska bärförmågan. I FE-beräkningen är materialet antaget ortotropiskt homogent, men det finns alltid svagheter i materialet även om provkropparna utvalts med stor omsorg. Den valda slumpmodellen (4.11) beskriver det slumpmässiga beteendet väl hos de dragbelastade provkropparna när uppmätt uppsprickningslast jämförs med beräknad. Detta framgår av residualanalysen och studium av fördelningsplottar. Emellertid är antalet provningar begränsat. Det går därför inte att styrka hypotesen att den skenbara brottsegheten är konstant med $0.3 < \alpha < 0.7$ eller att mätdata verkligen kommer från en lognormalfördelning. Å andra sidan förefaller det inte finnas någon invändning mot antagandet om lognormalfördelning. Variationskoefficienten för skattningen av beräkningskoefficienterna är mellan 15-20 % vilket är som förväntat för provning med trä.

7.2 Beräkningsuttryck i konventionella storheter

För att uttrycka det slutgiltiga beräkningsuttrycket i konventionella storheter antas att den skenbara brottsegheten \widetilde{K}_c är direkt proportionell mot skjuvhållfastheten τ_f . Eftersom provningen visar att det är osannolikt med simultan sprickbildning vid båda urtagen samt att skattat värde på \widetilde{K}_c överrensstämmer bäst med litteraturen när en spricka antas, används beräkningsuttrycket för en spricka (Fall 1) i detta avsnitt. För felfritt virke uppges karakteristisk skjuvhållfasthet $\tau_{f,k}$ vara lika med 4 MPa, om genomsnittlig densitet är 446.4 kg/m³ vid 12 % fuktkvot [14]. Korrelationskoefficienten mellan densitet och skjuvhållfasthet är 0.5 [24]. Eftersom medeldensiteten på de provade kropparna var 392.3 kg/m³ så blir skjuvhållfastheten för provkropparna

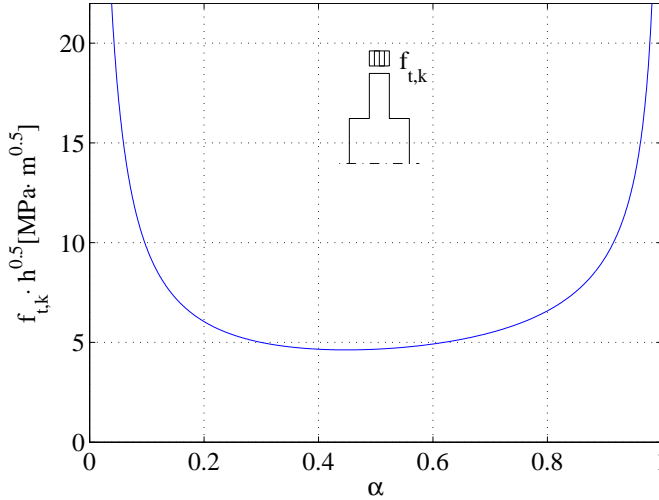
$$\tau_{f,k} = 4 \left(1 + 0.5 \frac{392.3 - 446.4}{446.4} \right) = 3.8 \text{ MPa}$$

Direktproportionaliteten mellan skenbar brottseghet och skjuvhållfasthet innebär att för någon konstant k gäller

$$\widetilde{K}_{c,k} = k \tau_{f,k}$$

Insättning av karakteristiskt värde på \widetilde{K}_c från Tabell 6.2 och $\tau_{f,k}$ ger

$$1.449 \cdot 10^6 = 0.38 \cdot 3.8 \cdot 10^6$$



Figur 7.1: Hållfasthet med avseende på uppsprickning $f_{t,k}$ multiplicerat med roten ur tvärsnittshöjden \sqrt{h} som funktion av α för fallet att $\beta = 0.5$ och $e_P = 0$.

det vill säga

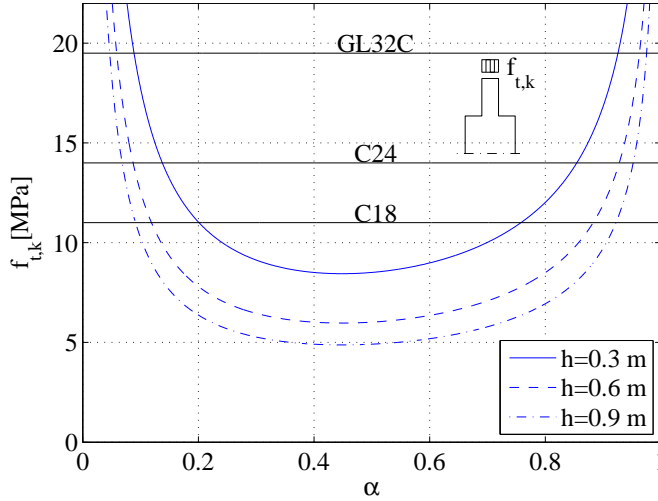
$$\widetilde{K}_{c,k} = 0.38\tau_{f,k}$$

om $\tau_{f,k}$ anges i Pa och $\widetilde{K}_{c,k}$ i $\text{N}/\text{m}^{\frac{3}{2}}$. För fallet med centrisk dragkraft och två likadana urtag kan bärförmågan (4.6) sidan 52 således skrivas

$$P_{c,k} = 0.38\tau_{f,k} \frac{b\sqrt{2h}}{\sqrt{-1 + \frac{2}{1+\alpha} + 6\frac{(1-\alpha)^2}{(1+\alpha)^3}}}$$

Där P anges i N samt b och h i m. Resultatet illustreras i Figur 7.1 där hållfastheten med avseende på uppsprickning multiplicerat med roten ur tvärsnittshöjden \sqrt{h} beskrivs som funktion av α . I Figur 7.2 visas hållfastheten med avseende på uppsprickning som en funktion av α för några givna värden på h .

I det allmänna fallet är $\widetilde{K}_{c,m} = \sqrt{E_m G_c}$ där $G_c = 283r_I + 1240(1-r_I)$. Sambandet mellan E_m och E_k är enligt EN338 [16] $E_k = \frac{2}{3}E_m$. Andelen Mod I sätts förenklat till 0.6 vilket ger att $G_c = 670 \text{ Nm}/\text{m}^2$. Figur 6.12 visar hur lite andelen Mod I avviker från 0.6 då $\alpha \gtrsim 0.8$ oberoende av β och e_P . Då $\alpha \leq 0.8$ är förenklingen på säkra sidan. Med $G_c = 670 \text{ Nm}/\text{m}^2$ samt beräkningsfaktorn 0.56 från avsnitt 6.1.6 insatt i (4.5) sidan 52 fås den slutgiltiga allmänna formeln till



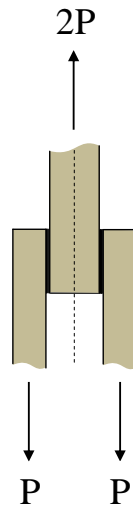
Figur 7.2: Hållfasthet med avseende på uppsprickning $f_{t,k}$ som funktion av α för några värden på tvärsnittshöjden h i fallet att $\beta = 0.5$ och $e_P = 0$. Som jämförelse har karakteristisk draghållfasthet markerats för några limträ- och virkeskvaliteter.

$$P_{c,k} = 22 \frac{b\sqrt{2hE_k}}{\sqrt{-1 + \frac{1}{\alpha+\beta-\alpha\beta} + \frac{12(e_p+(\beta-\alpha\beta)h/2)^2}{h^2(\alpha+\beta-\alpha\beta)^3} - \frac{12(e_p+(-1+\alpha+2\beta-2\alpha\beta)h/2)^2}{h^2}}} \quad (7.1)$$

Med $E_k = 7400$ MPa (C24) och $e_P = 0$ ger (7.1) bärförmågan med avseende på uppsprickning som funktion av värdet på α . Sambandet illustreras för några värden på β i Figur 7.4 där y-axeln visar hållfastheten med avseende på uppsprickning multiplicerat med roten ur tvärsnittshöjden $f_{c,t}\sqrt{h}$. Exempel på kurvornas utseende för givna värden på h ges i Figurerna 7.6, 7.7 och 7.8 i vilka $h = 0.3, 0.6$ respektive 0.9 . Figur 7.5 skiljer sig från Figur 7.4 genom att dragkraften har en excentricitet om $e_P = 0.025h$.

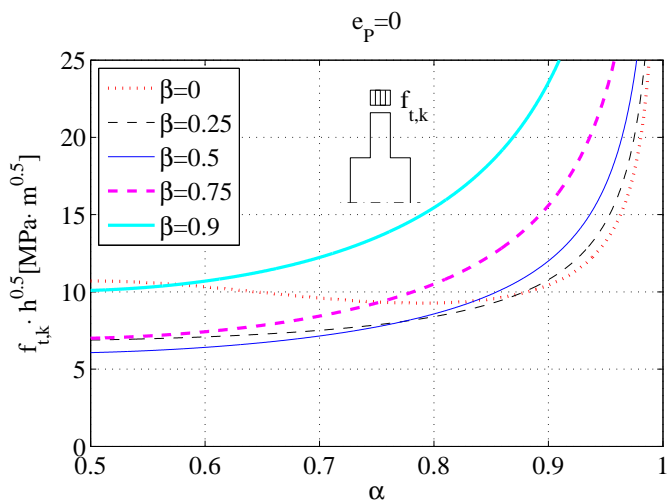
(7.1) kan alternativt uttryckas i $\tau_{f,k}$ istället för i G_c och i E_k . Detta görs genom att $22E_k$ ersätts med $0.38\tau_{f,k}$. Det är visserligen inte i linje med den valda metoden (jämför med schemat i Figur 5.1) men det ger ett snarlikt resultat.

Hur beräkningsuttrycken kan användas i praktiken visas här med ett exempel som bygger på ett av fallen vilket hänvisades till i avsnitt 2.1 sidan 8:

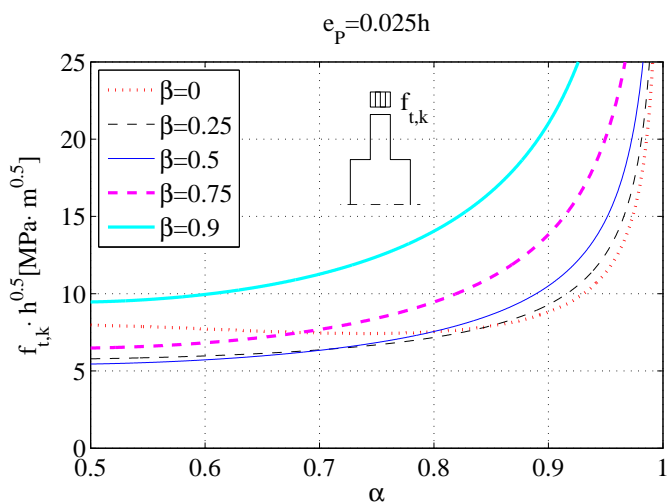


Figur 7.3: Dragbelastat tvåskärigt limträförband med symmetrilinje.

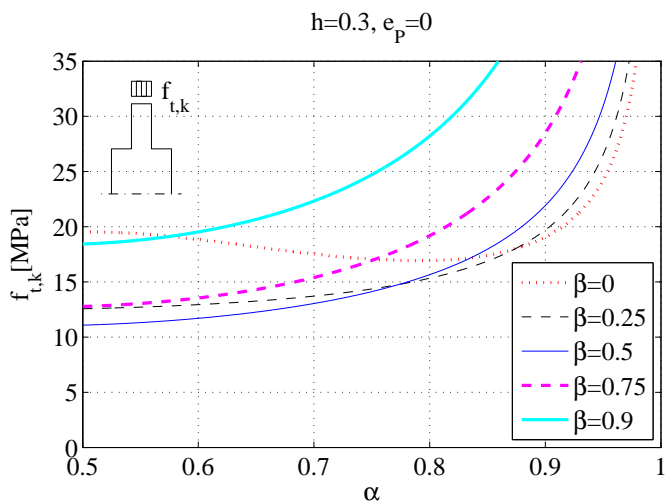
Exempel 7.2.1 En sträva av limträ med tvärsnittshöjden 300 mm ges två likadana urtag som lämnar en reducerad tvärsnitthöjd om 215 mm. Detta ger med beteckningen i Figur 3.2a att $\alpha \approx 0.7$. Strävan ingår i en bågkonstruktion. Med diagrammet i Figur 7.2 uppskattas hållfastheten med avseende på uppsprickning till 10 MPa vilket är ungefär hälften av draghållfastheten för GL32C.



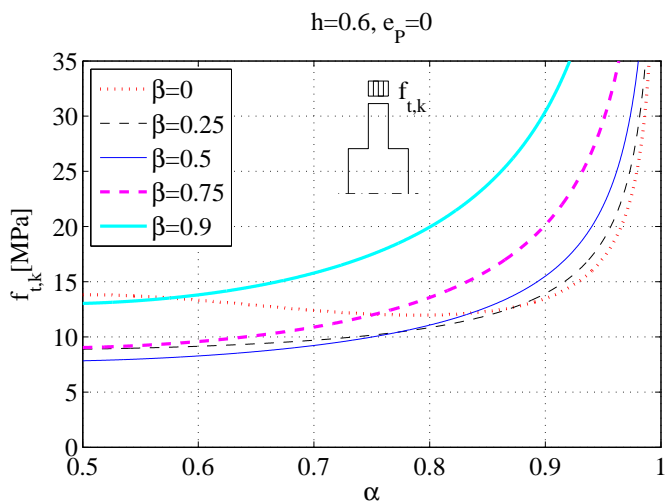
Figur 7.4: Hållfasthet med avseende på uppsprickning $f_{t,k}$ multiplicerad med roten ur tvärsnittshöjden \sqrt{h} som funktion av α för $e_p = 0$ och några värden på β . Andelen Mod I är 0.6.



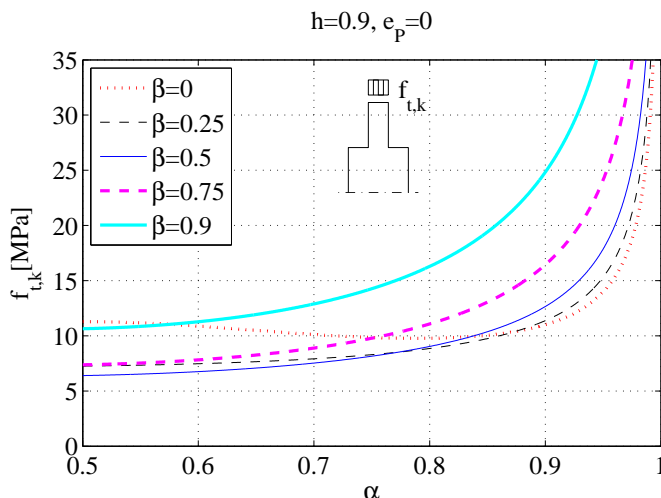
Figur 7.5: Hållfasthet med avseende på uppsprickning $f_{t,k}$ multiplicerad med roten ur tvärsnittshöjden \sqrt{h} som funktion av α för $e_p = 0.025h$ och några värden på β . Andelen Mod I är 0.6.



Figur 7.6: Hållfasthet med avseende på uppsprickning $f_{t,k}$ som funktion av α för $e_p = 0$, $h = 0.3$ m och några värden på β . Andelen Mod I är 0.6.



Figur 7.7: Hållfasthet med avseende på uppsprickning $f_{t,k}$ som funktion av α för $e_p = 0$, $h = 0.6$ m och några värden på β . Andelen Mod I är 0.6.



Figur 7.8: Hållfasthet med avseende på uppsprickning $f_{t,k}$ som funktion av α för $e_p = 0$, $h = 0.9$ m och några värden på β . Andelen Mod I är 0.6.

7.3 Förslag på vidare forskning

- Storlekseffekten är viktig att utreda och kan verifieras genom fler provningar av stora träkroppar.
- Verifiera beräkningsuttrycket för limträ genom provning på limträ.
- Verifiera beräkningsuttrycket med avseende på lastexcentricitet genom provning.
- Utreda och verifiera effekten av sektionssändringar med generell form, till exempel avfasade urtag.
- Utreda om det går att förutsäga hur sprickan vid urtaget av en limträkonstruktion propagerar med avseende på riktning.
- Utreda verkan av långtidsbelastning och av uttorkningsprickor.

7.4 Sammanfattning

Med utgångspunkt i 1) första ordningens linjärelastiska teori, 2) en förenklad materialmodell med antagande om homogenitet och 3) ett brottmekaniskt villkor på den frigjorda töjningsenergin vid uppsprickning kunde det härledas ett beräkningsuttryck för bärförmågan med

avseende på uppsprickning i en dragbelastad träkropp med urtag. För centriskt dragbelastade träkroppar med två likadana urtag verifierades beräkningsuttrycket genom provning av 21 stycken träkroppar. De slumpmässiga avvikelserna modellerades med en logaritmisk normalfördelning och karakteristiskt värde på bärförmågan beräknades som 5-percentilen med 75 % konfidensgrad. Beräkningskoefficienten är i detta fall den skenbara brottsegheten \widetilde{K}_c . Med antagande om direkt proportionalitet mellan skenbar brottseghet och skjuvhållfasthet gavs beräkningsuttrycket i konventionella storheter, se vidare rutan nedan under rubriken Beräkningsuttryck 1. Simulering med FEM och noduppsläppning bekräftade antagandet att \widetilde{K}_c i princip inte varierar med urtagsdjupet för $0.3 \leq \alpha \leq 0.7$. Emellertid behöver storlekseffekten utredas då provningen endast omfattar små träkroppar.

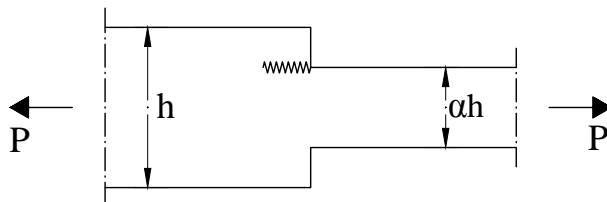
I det allmänna fallet med en (excentriskt) dragbelastad träkropp med ett eller två (inte nödvändigtvis lika djupa) urtag verifierades beräkningsuttrycket genom simulering med FEM och noduppsläppningsmetod. Materialmodellen bygger på antagande om ortotropi. Den numeriska beräkningsmodellen kalibrerades först mot provningsresultaten med en beräkningskoefficient $D = 0.5811$ (karakteristiskt värde). Resultaten visar att beräkningsuttrycket är giltigt även för mer generella geometrier. Vidare bygger uttrycket på förenklingen att andelen Mod I i modblandningen är konstant lika med 60 %. Antagandet ligger på säkra sidan. Det är framför allt för små till medelstora värden på α som bärförmågan riskerar att underskattas på grund av förenklingen. Beräkningsuttrycket återges nedan i rutan under rubriken Beräkningsuttryck 2.

Den numeriska beräkningsmodellen som bygger på linjärelastisk brottmekanik visade sig vara nära nog lika bra på att förutsäga bärförmågan hos de provade träkropparna som den mer raffinerade modellen med olinjära fjädrar och kohesiv zon.

Beräkningsuttryck 1 För en centriskt dragbelastad kropp med två likadana urtag gäller

$$P_{c,k} = 0.38\tau_{f,k} \frac{b\sqrt{2h}}{\sqrt{-1 + \frac{2}{1+\alpha} + 6\frac{(1-\alpha)^2}{(1+\alpha)^3}}}$$

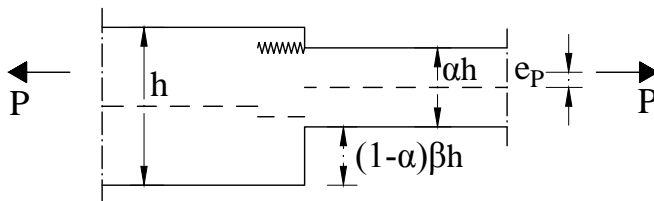
där $P_{c,k}$ anges i [N], $\tau_{f,k}$ i [Pa] samt b och h i [m].



Beräkningsuttryck 2 För en dragbelastad kropp med ett eller två urtag gäller

$$P_{c,k} = 22 \frac{b\sqrt{2hE_k}}{\sqrt{-1 + \frac{1}{\alpha+\beta-\alpha\beta} + \frac{12(e_p+(\beta-\alpha\beta)h/2)^2}{h^2(\alpha+\beta-\alpha\beta)^3} - \frac{12(e_p+(-1+\alpha+2\beta-2\alpha\beta)h/2)^2}{h^2}}}$$

där $P_{c,k}$ anges i [N], E_k i [Pa] samt b , e_p och h i [m].



Källor

- [1] ABAQUS FEA, Version 6.10.1. Dassault Systèmes Simulia Corp
- [2] BBC Classical Serial: *No highway* Hämtad 2013-12-23
<http://www.bbc.co.uk/programmes/b00tf9nr>
- [3] Berbom Dahl Kristian (2009): *Mechanical properties of clear wood from Norway spruce* Doktorsavhandling, Avdelningen för konstruktionsteknik, Norges teknisk- och naturvetenskapliga universitet, Norge
- [4] Bibel, George (2008): *Beyond the black box. The forensics of airplane crashes* JHU Press
- [5] Blass, Hans J. (red.) (1995): *Timber engineering. Step 1 : basis of design, material properties, structural components and joints* Salland De Lange, Deventer. Nederländerna.
- [6] Blom, Gunnar (1984): *Sannolikhets teori med tillämpningar*. Studentlitteratur, Lund
- [7] Blom, Gunnar och Holmquist, Göran (1998): *Statistik teori med tillämpningar*. Studentlitteratur, Lund
- [8] Boverket (1994): *Dimensionering genom provning*. Upplaga 1:1, Emmaboda
- [9] CERN Bulletin (2006-07-31): "Python : the holy grail of programming". *CERN Publications*. Hämtad 2013-12-15.
<http://cds.cern.ch/journal/CERNBulletin/2006/31/News%20Articles/974627?ln=en>
- [10] Chakraborti, S och LiSource, J (November 2007): *The American Statistician*, "Confidence Interval Estimation of a Normal Percentile" Vol. 61, Nr. 4

- [11] Danielsson, Henrik (2013): *Perpendicular to grain fracture analysis of wooden structural elements. Models and applications*. Doktorsavhandling, Rapport TVSM-1024, Avdelningen för byggnadsmekanik, Lunds Universitet
- [12] Dassault Systèmes Simulia Corp (2014): “ABAQUS FEA”, *ABAQUS Portfolio*. Hämtad 2014-01-05 <http://www.3ds.com/products-services/simulia/portfolio/abaqus/abaqus-portfolio/>
- [13] Dassault Systèmes Simulia Corp (2008): *ABAQUS Analysis user’s manual version 6.9*
- [14] Densler, J. K. & Glos, P (2005): *Determination of shear strength values according to EN 408*, Holzforschung, Technische Universität, Munchen, Tyskland.
- [15] SS-EN 1995-1-1:2004: Dimensionering av träkonstruktioner – Del 1-1: Allmänt – Gemensamma regler och regler för byggnader
- [16] EN 338:2009 Structural timber – Strength classes
- [17] ENV 1991-1 (Oktober 1993): *Eurocode 1 - Basis of design and action on structures. Part 1: Basis of design*.
- [18] “Organizations using Python”. *Python Software Foundation*. Hämtad 2013-12-15. <https://wiki.python.org/moin/OrganizationsUsingPython>
- [19] “General Python FAQ”. Python v2.7.6 documentation. Hämtad 2013-12-15. <http://docs.python.org/2/faq/general.html#why-is-it-called-python>
- [20] Grahn, Ragnar och Jansson, Per-Åke (2002): *Mekanik, statik och dynamik*. Studentlitteratur
- [21] Gustafsson, Per-Johan, m. fl (1991): *Application of fracture mechanics to timber structures RILEM state-of-the-art report*. Statens tekniska forskningscentral i Finland
- [22] Gustafsson, Per-Johan (1988): *A study of strength of notched beams*. International council for building research studies and documentation. Working commission W18A - Timber structures.

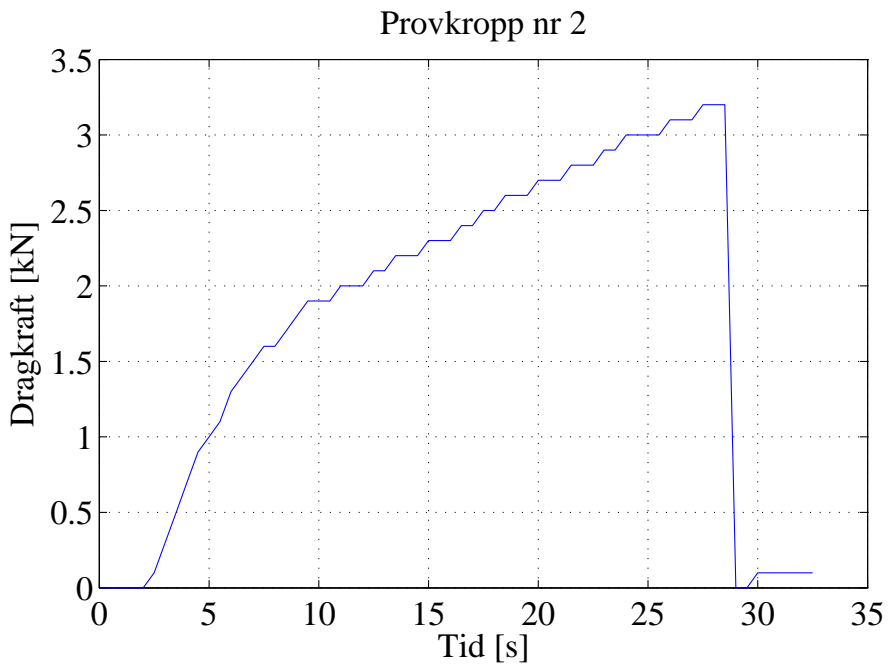
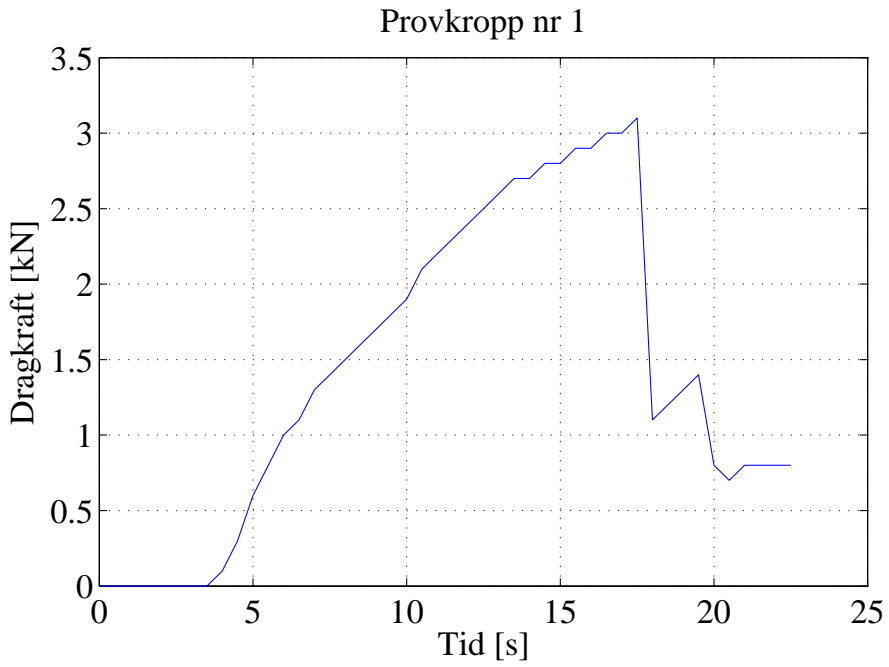
- [23] Hellan, Kåre (1984): *Introduction to fracture mechanics*. McGraw-Hill Book Company
- [24] Isaksson, Tord & Mårtensson, Annika & Thelandersson, Sven (2010): *Byggkonstruktion*. Studentlitteratur AB, Lund
- [25] Isaksson, Tord (1999): *Modelling the variability of bending strength in structural timber. Length and load configuration effects*. Doktorsavhandling, Rapport TVBK-1015, Avdelningen för konstruktions teknik, Lunds Universitet
- [26] Jalote, Pankaj (2005): *A concise introduction to software engineering*. Springer Verlag London Limited.
- [27] Karlsson, Bengt (1995): *Folkets Park i Lund - En hundraårig historia 1895 - 1995*, BTJ Tryck AB, Lund
- [28] Landin, P. J. (Mars 1966): "The next 700 programming languages". *Communications of the ACM* 9 (3): 157-166. Hämtad 2013-12-15. <http://fsl.cs.illinois.edu/images/e/ef/P157-landin.pdf>
- [29] Larsen, Hans Jørgen & Riberholt, H. (1972): *Forsøg med Uklassificeret Konstruktionstræ* Danmarks Tekniske Højskole, Danmark
- [30] Lawless, Jerald F (1982): *Statistical models and methods for lifetime data*. John Wiley & Sons. Inc
- [31] MacSweeney, Greg (17 maj 2005): "Dassault Systèmes Acquires Realistic Simulation Software Firm", *Managing Automation*. Hämtad 2014-01-05 http://www.managingautomation.com/maonline/news/read/Dassault_Systèmes_Acquires_Realistic_Simulation_Software_Firm_6934
- [32] Martelli, Alex: *Python Cookbook* 2:a uppl., sid. 230
- [33] MATLAB, Version R2012b. The MathWorks Inc.
- [34] Modéer, Matz (1980): *Elementär brottmekanik*. Rapport TVBM-3006, Avdelningen för byggnadsmateriallära, Lunds Universitet
- [35] National Transportation Safety Board: *Aloha Airlines Boeing 737-297 at Kahului Airport, Hawaii* Bildkälla hämtad 2013-12-28. <http://aviation-safety.net/photos/displayphoto.php?id=19880428-0&vnr=2&kind=C>

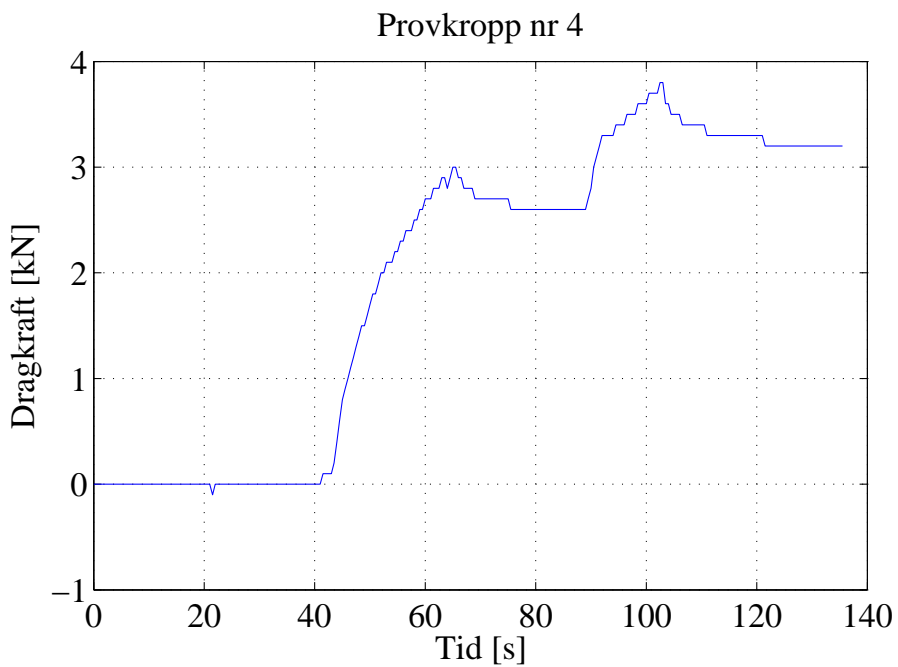
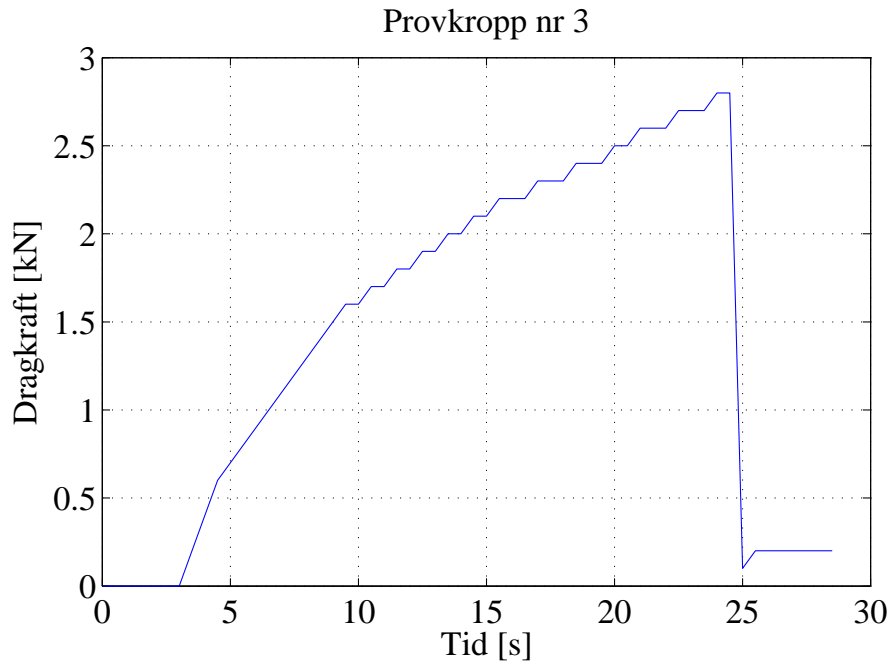
- [36] Nationalencyklopedin. Uppslagsord: Comet. Nätupplagan. 2013-10-25
- [37] Olbjer, Lennar (2000): *Experimentell och industriell statistik* Femte upplagan, Lunds tekniska högskola, Lund
- [38] Ottosen, N & Petersson, H (1992): *Introduction to the Finite Element Method* Prentice Hall, Storbritannien
- [39] Petersson, Per-Erik (1981): *Crack growth and development of fracture zones in plain concrete and similar materials*. Doktorsavhandling, Rapport TVBM-1006, Avdelningen för byggnadsmateriallära, Lunds Universitet
- [40] Reed, R. P., m. fl (1983): *The economic effects of fracture in the United States of America*. Washington, D.C: U.S. Dept. of Commerce, National Bureau of Standards
- [41] Statens Byggeforskningsinstitut (1983): *Traekonstruktioner. Beräkning*. SBI-Anvisning 135, Glostrup, Danmark.
- [42] Rhode Island Science and Technology Advisory Council (2012): *David Hibbitt, Ph.D. - Founder and Former Chairman, ABAQUS, Inc.* Hämtad 2014-01-05 <http://stac.ri.gov/about/council/david-hibbitt/>
- [43] van Rossum, G (2009-01-20): "A brief timeline of Python". *The history of Python*. Hämtad 2013-12-15 <http://python-history.blogspot.se/2009/01/brief-timeline-of-python.html>
- [44] Roylance, David (1995): *Mechanics of materials*. Wiley
- [45] Shafer, Daniel G. (17 January 2003): "Python Streamlines Space Shuttle Mission Design". *Python Software Foundation*. Hämtad 2013-12-15. <http://www.python.org/about/success/usa/>
- [46] Stefansson, F (2001): *Fracture analysis of orthotropic beams - Linear elastic and non-linear methods*. Licentiatavhandling, Rapport TVSM-3029. Avdelningen för byggnadsmekanik, Lunds Universitet.
- [47] Sveriges provnings- och forskningsinstitut: *Dimensionering av träkonstruktioner genom provning*. SP Rapport 1994:60

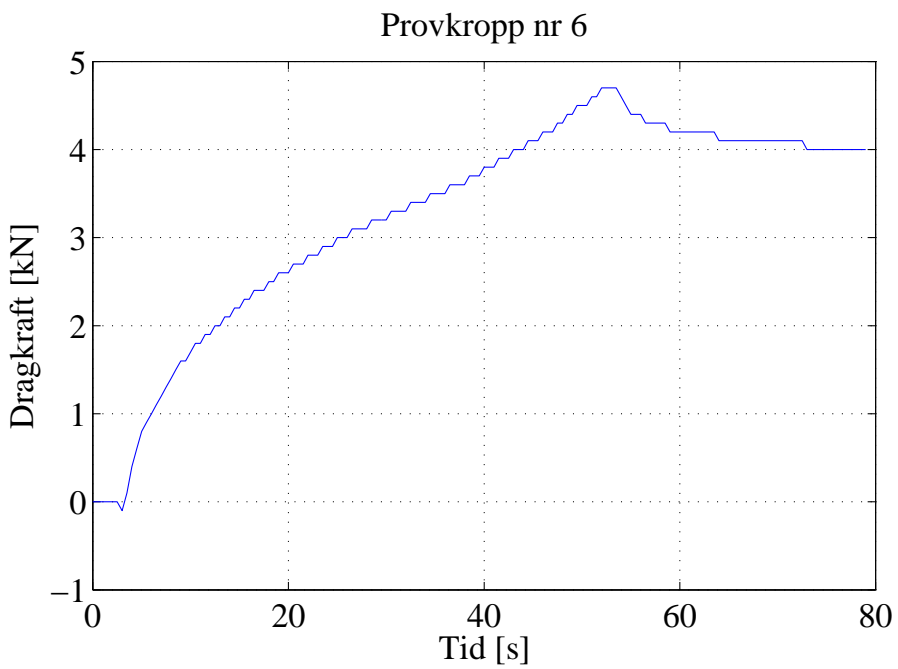
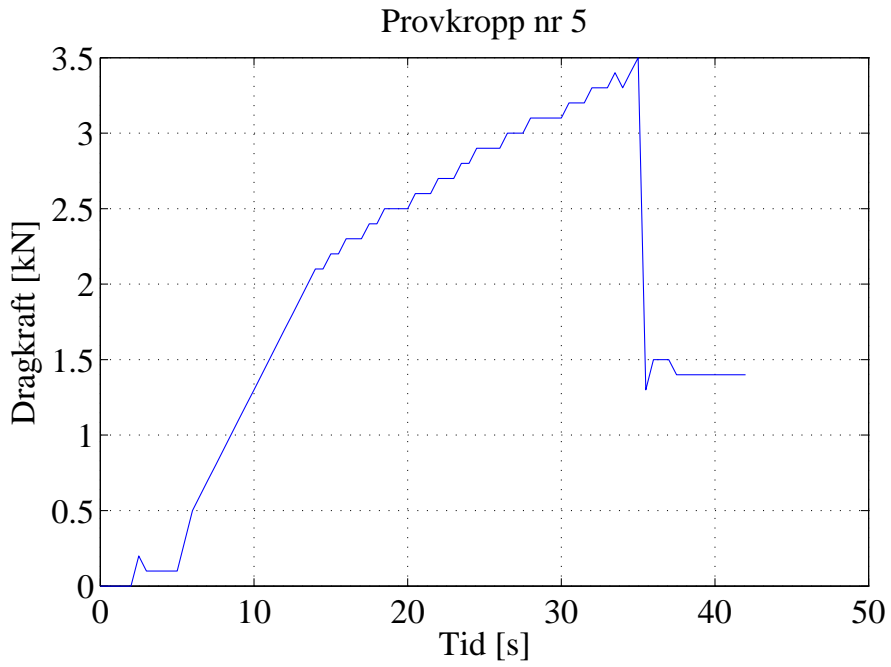
- [48] Swedish Wood (2011): *Design of timber structures* Exaktaprinting, Malmö
- [49] Thompson, Peter (2001): "How Much Did the Liberty Shipbuilders Learn? New Evidence for an Old Case Study". *The Journal of Political Economy*. The University of Chicago Press
- [50] TIOBE Software Index (2013): "TIOBE Programming Community Index Python". Hämtad 2013-12-15. <http://www.tiobe.com/index.php/paperinfo/tpci/Python.html>
- [51] Tegel, Willy m. fl. (19 december 2012): *Early Neolithic Water Wells Reveal the World's Oldest Wood Architecture* DOI: 10.1371/journal.pone.0051374
- [52] Uhre Pedersen, Martin (2002): *Dowel type timber connections. Strength modelling*. Doktorsavhandling, Rapport BYG DTU R-039, Danmarks tekniska universitet, Danmark
- [53] Vännman, Kerstin (2002): *Matematisk statistik*. Studentlitteratur, Lund
- [54] Wernersson, Håkan (1990): *Wood adhesive bonds-Fracture softening properties in shear and in tension* Rapport TVSM-3012, Avdelningen för byggnadsmekanik, Lunds Universitet.
- [55] Wikipedia Commons: *Aeromaritime de Havilland DH-106 Comet* Bildkälla hämtad 2013-12-28. http://commons.wikimedia.org/wiki/File:Aeromaritime_Comet_Groves.jpg
- [56] Wikipedia Commons: *SS Schenectady* Bildkälla hämtad 2013-12-28. <http://en.wikipedia.org/wiki/File:TankerSchenectady.jpg>
- [57] Wright, Douglas: *Fracture - some maritime examples* Mechanical Engineering Department, University of Western Australia. Hämtad 2013-12-28. <http://school.mech.uwa.edu.au/~dwright/DANotes/fracture/maritime/maritime.html>
- [58] *SS John P Gaines* Bildkälla hämtad 2013-12-28. <http://school.mech.uwa.edu.au/~dwright/DANotes/fracture/maritime/JohnPGaines.jpeg>

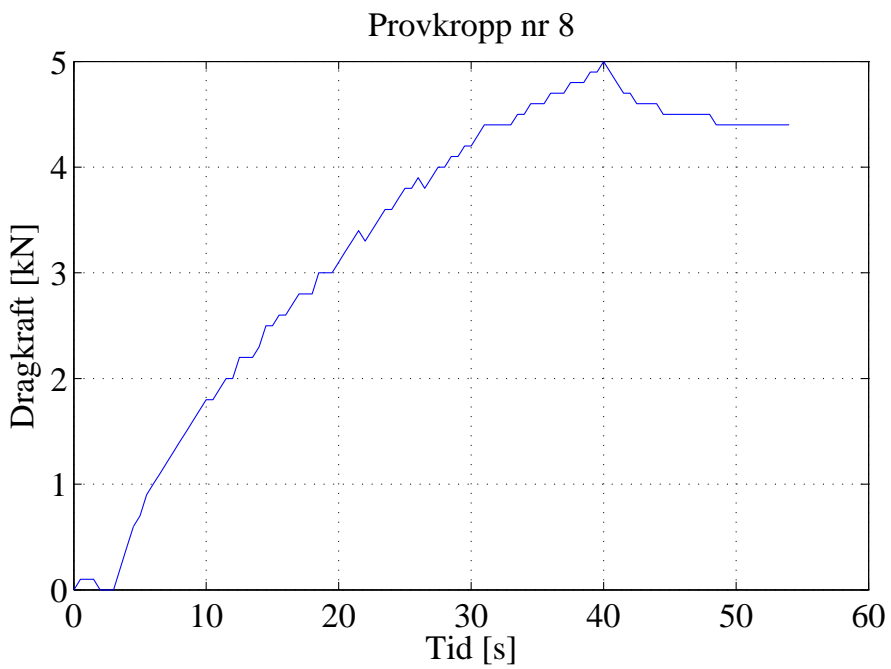
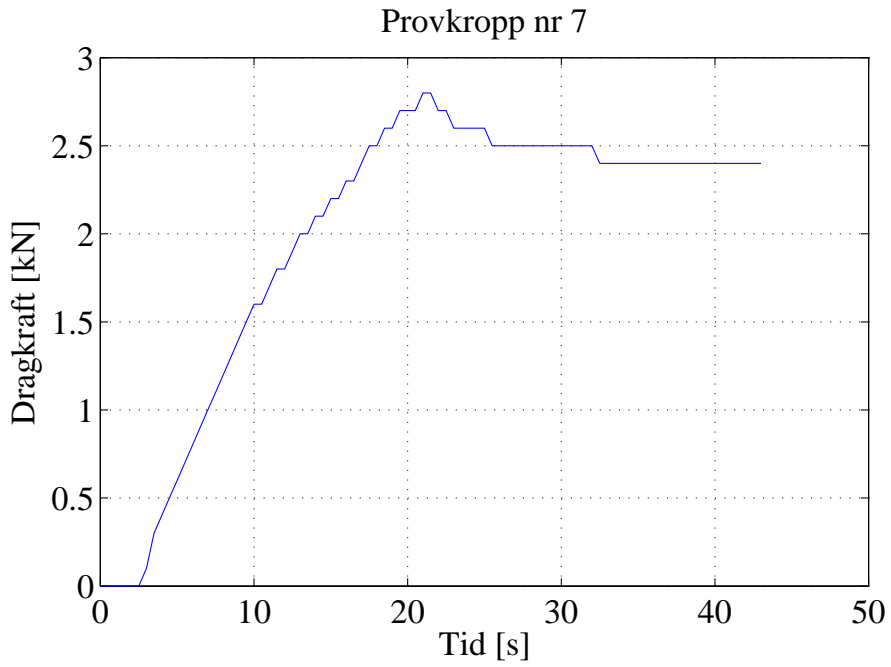
Bilaga 1

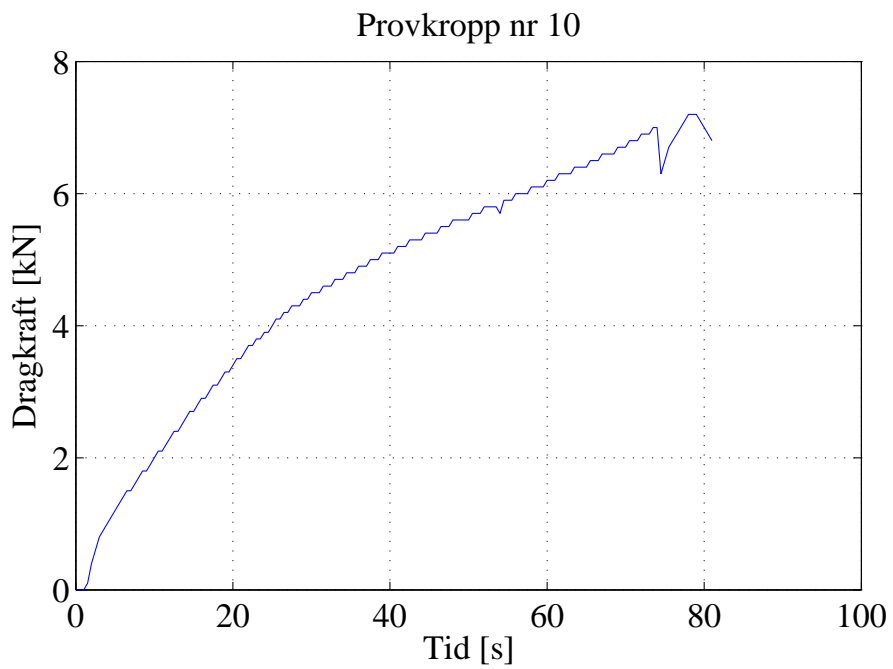
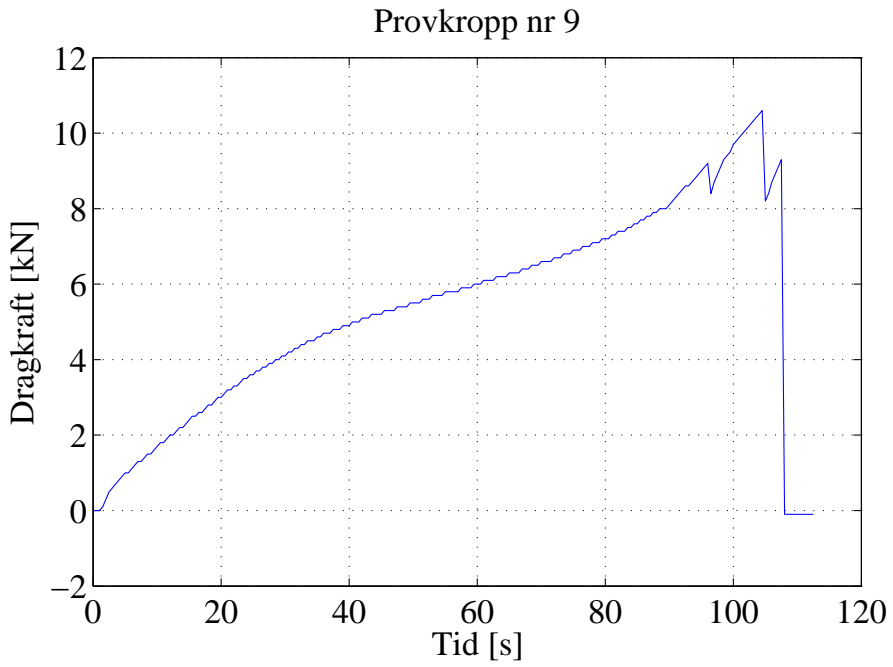
Last-tid-diagram från provningen. Numrering enligt Figur 6.1 på sidan 78.

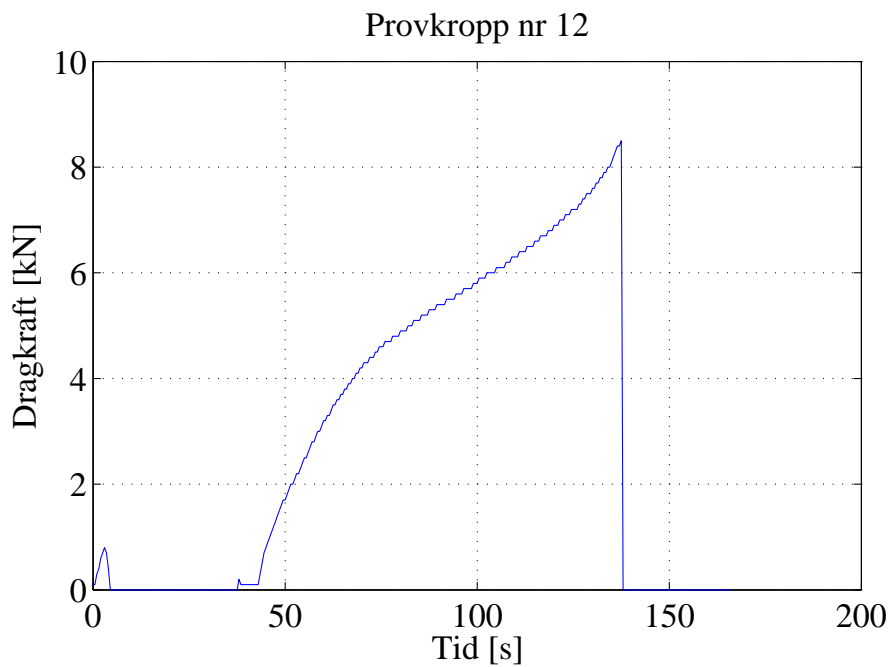
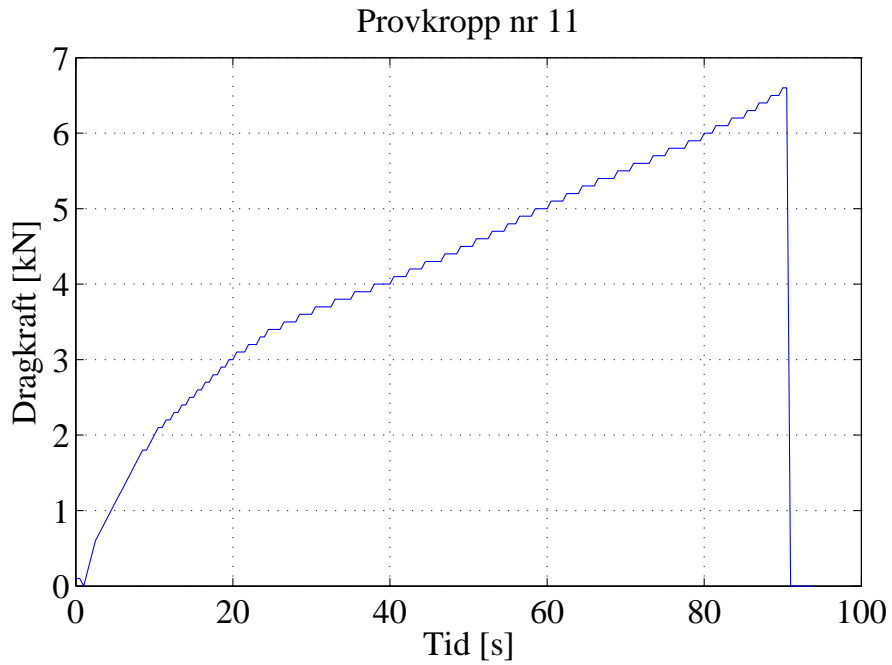


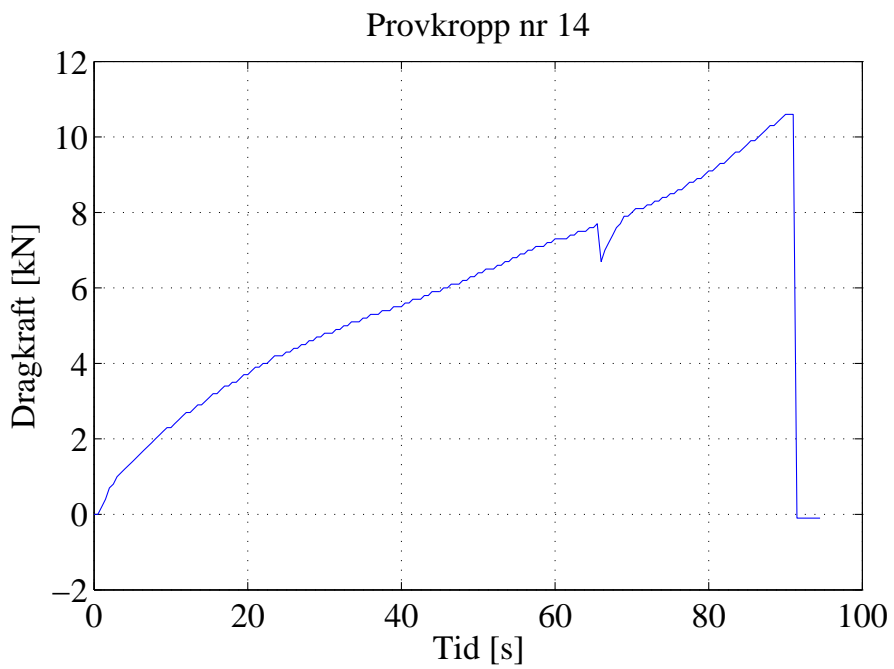
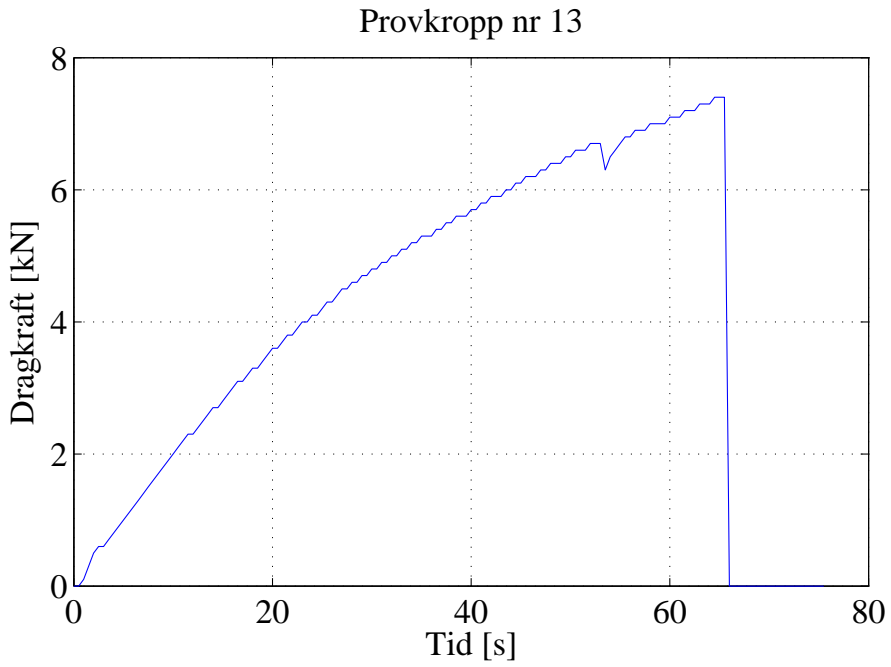


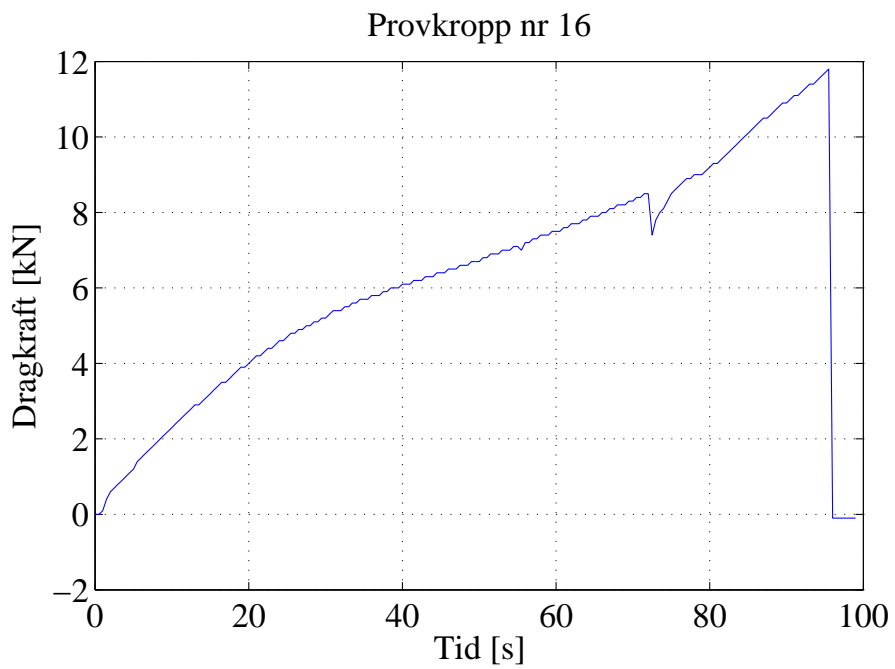
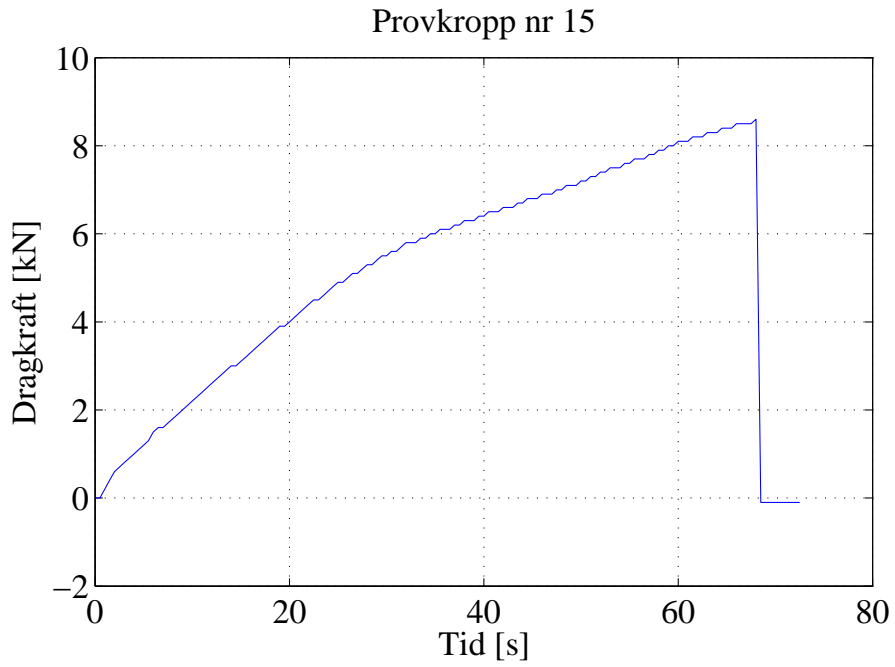


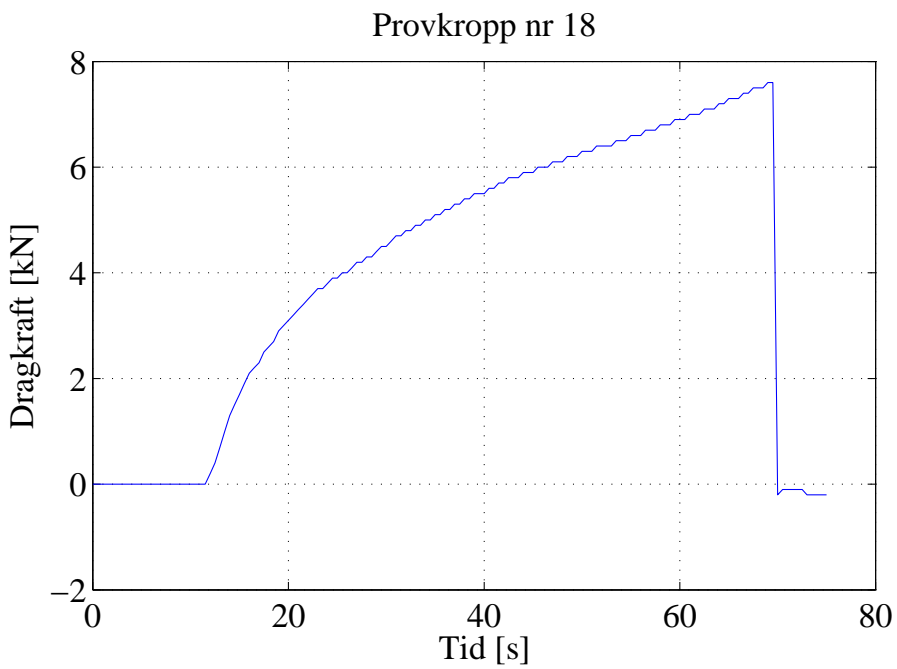
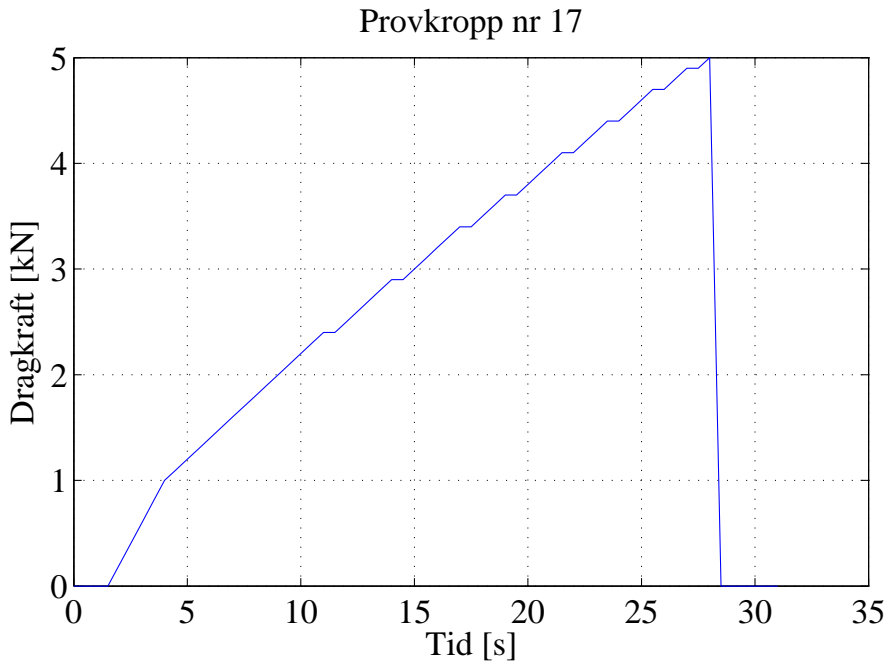


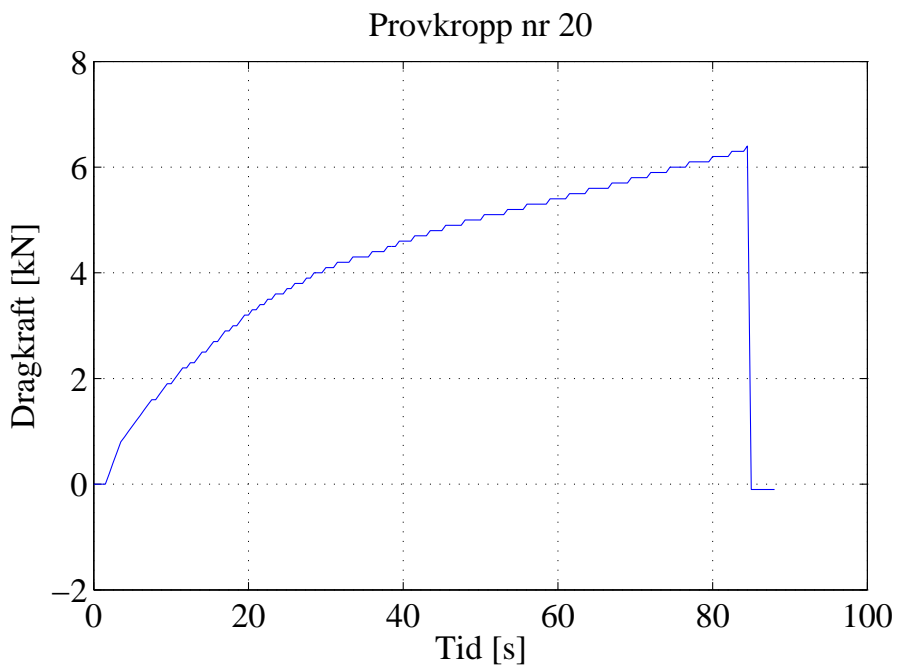
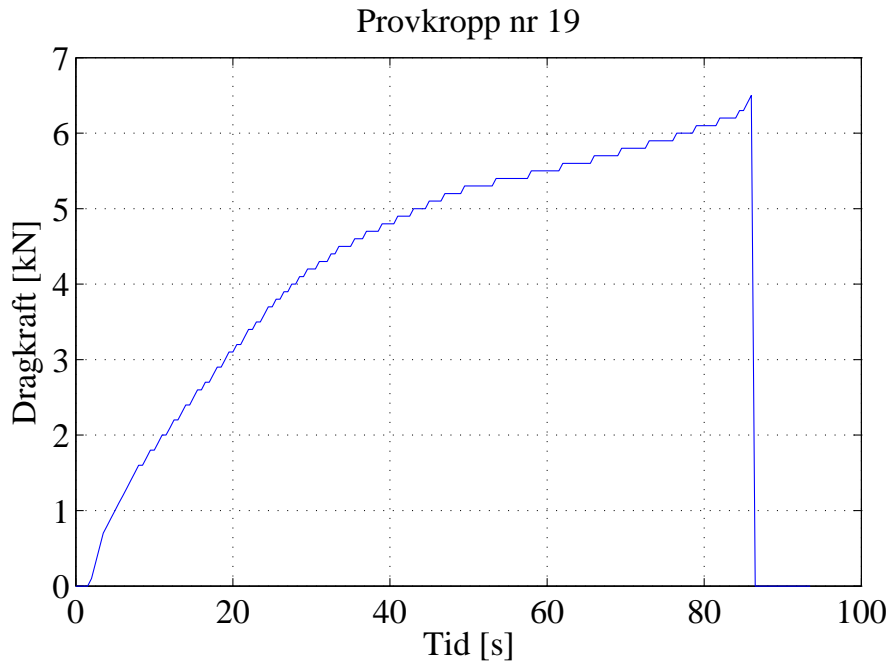


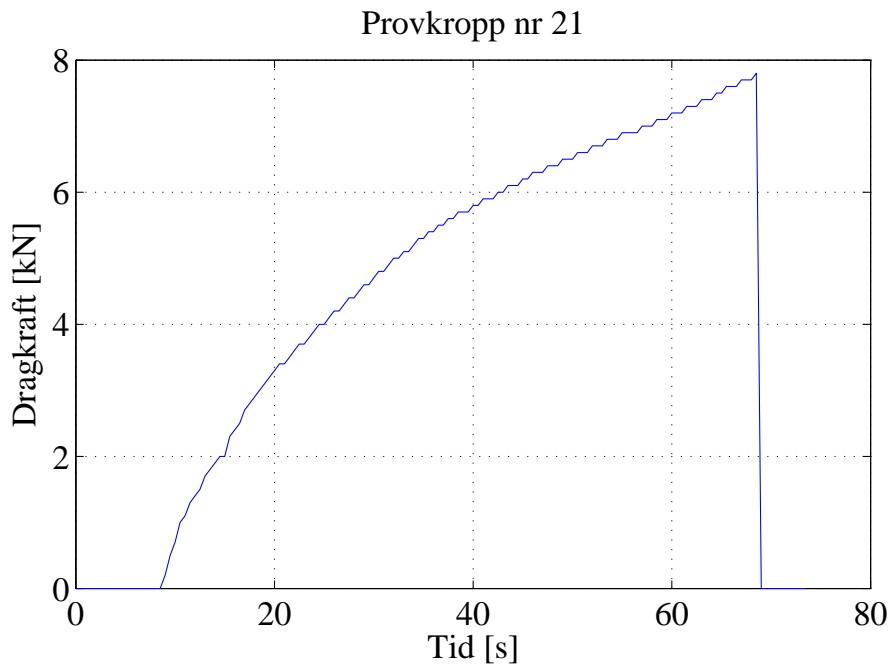












Bilaga 2

Finita elementmetoden

Introduktion

Många ingenjörsmässiga problem modelleras med hjälp av differentialekvationer. I allmänhet är det inte möjligt att hitta explicita lösningsuttryck som är praktiska för handberäkningar. Finita elementmetoden är en metod för att hitta approximativa lösningar till differentialekvationer genom indelning i mindre komplexa delar. I praktiska tillämpningar blir antalet ekvationer ofta stort och metoden olämplig för handberäkning. Däremot lämpar den sig väl för datorberäkning. En fördel är att komplicerade geometrier kan modelleras genom indelningen i finita element. Genom att approximera lösningen med en funktion som varierar kontinuerligt över elementgränserna och som är åtminstone en gång differentierbar uppfylls kravet på konvergens [38]. Vilken noggrannhet som fås i lösningen beror dels på hur elementindelningen görs och hur randvillkoren definieras. Hur lång tid det tar att generera lösningen beror på vilken datorkraft som står till förfogande. En finita elementmodell kan vara bra eller dålig beroende på hur väl den efterliknar verkligheten som undersöks. En lösning är aldrig så god som modellen själv. Den följande framställningen bygger på [11] och [38] och representerar inte författarnas eget resultat.

Standardmetoden

Betrakta en kropp med volymen V , densiteten ρ , ytan S , ytlasten \mathbf{t} och volymlasten \mathbf{b} såsom illustreras i Figur 7.9. Om \mathbf{u} är förskjutningsvektorn så kan Newtons andra lag uppställas [11]

$$\int_S \mathbf{t} \, dS + \int_V \mathbf{b} \, dV = \int_V \rho \ddot{\mathbf{u}} \, dV \quad (7.2)$$

Statisk kraftjämvikt innebär att högerledet i (7.2) är lika med nollvektorn:

$$\int_S \mathbf{t} \, dS + \int_V \mathbf{b} \, dV = \mathbf{0} \quad (7.3)$$

Inför ett rätvinkligt koordinatsystem. Då är:

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

Med hjälp av Gauss divergenssats kan jämviktsekvationen (7.3) omskrivas [11]:

$$\int_V (\tilde{\nabla}^T \boldsymbol{\sigma} + \mathbf{b}) \, dV = \mathbf{0} \quad (7.4)$$

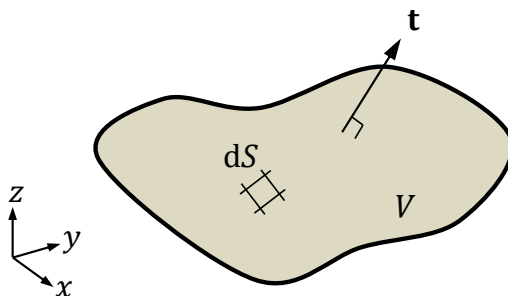
Här betecknar $\boldsymbol{\sigma}$ spänningsvektorn och $\tilde{\nabla}$ en differentialoperator enligt

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{bmatrix}, \quad \tilde{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix}$$

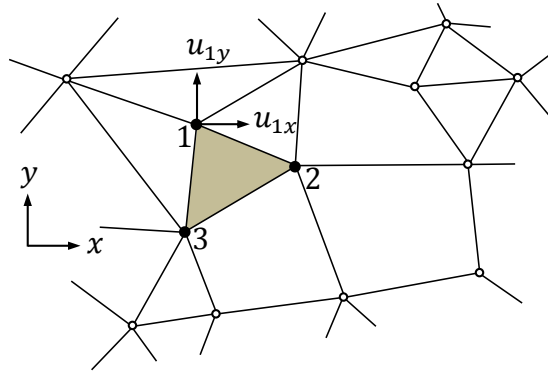
Eftersom kroppens volym från början var godtycklig ges av (7.4) det som brukar kallas den starka formen av jämviktsekvationen:

$$\tilde{\nabla}^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \quad (7.5)$$

Idén i det följande är att uttrycka $\boldsymbol{\sigma}$ i förskjutningarna \mathbf{u} med hjälp av ett konstitutivt samband (till exempel Hookes lag) och sedan approximera



Figur 7.9: Frilagd kropp med volymen V , densiteten ρ , ytan S , snittkraften \mathbf{t} och inre kraften \mathbf{b}



Figur 7.10: Illustration över ett finita elementnät med noder

\mathbf{u} på lämpligt vis. Emellertid inses att denna approximation måste vara åtminstone en gång differentierbar. Kravet sänks genom att skriva om jämviktsekvationen ytterligare. För effektiviteten i de numeriska beräkningarna är omskrivningen dessutom en fördel. Multiplicera därför den starka formen (7.5) med en godtycklig vektor \mathbf{v} , integrera över volymen V och tillämpa Gauss divergenssats en gång till för att erhålla den så kallade svaga formen av jämviktsekvationen

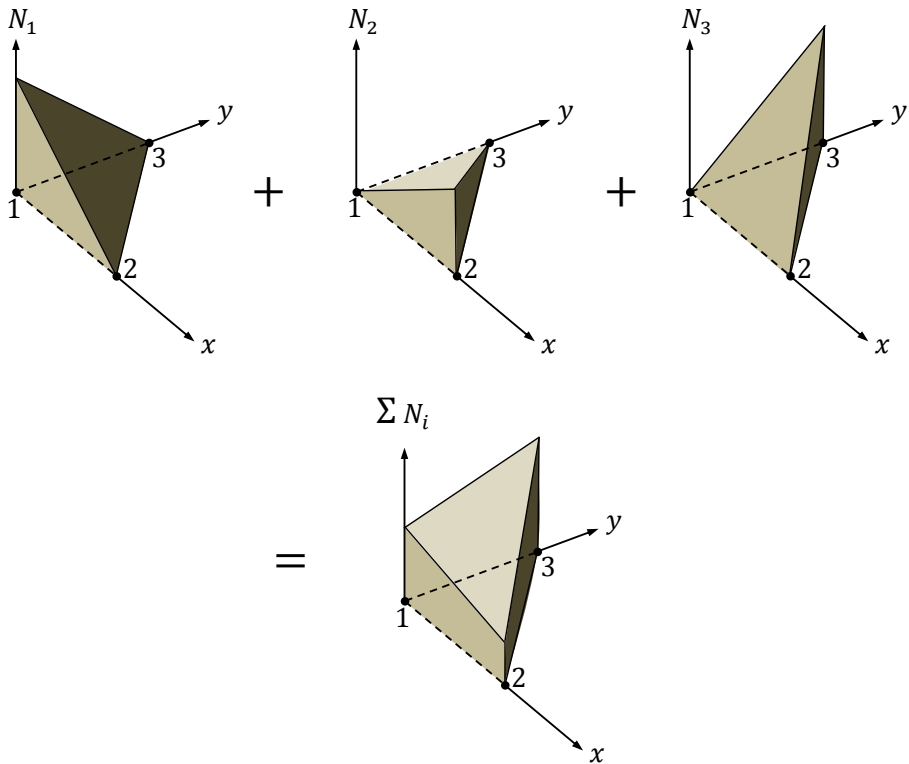
$$\int_V (\tilde{\nabla} \mathbf{v})^T \boldsymbol{\sigma} \, dV = \int_S \mathbf{v}^T \mathbf{t} \, dS + \int_V \mathbf{v}^T \mathbf{b} \, dV, \quad \mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (7.6)$$

Inför ett elementnät med noder i princip enligt Figur 7.10. Om nodförskjutningarna för ett element \mathbf{a}^e är kända kan också förskjutningarna mellan noderna elementet uppskattas genom exempelvis linjär viktning av nodvärdena, jämför Figur 7.11. Med hjälp av så kallade formfunktioner \mathbf{N}^e kan detta uttryckas som [38]:

$$\mathbf{u}^e \approx \mathbf{N}^e \mathbf{a}^e$$

För konvergens fordras att formfunktionerna \mathbf{N}^e är åtminstone en gång differentierbara och att de varierar kontinuerligt över elementgränserna. För sådana formfunktioner \mathbf{N}^e kan förskjutningarna beskrivas med en differentierbar funktion med utgångspunkt i elementnodernas förskjutningar och det gäller approximativt att

$$\mathbf{u} \approx \mathbf{N} \mathbf{a}$$



Figur 7.11: Med hjälp av formfunktioner kan variationen inom ett finita element uppskattas genom linjär viktning av nodvärdena.

Om $\mathbf{B} = \tilde{\nabla}\mathbf{N}$ fås töjningarna

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a} \quad (7.7)$$

Enligt Galerkins metod ansätts [38]

$$\mathbf{v} = \mathbf{N}\mathbf{c}$$

där det faktum att \mathbf{v} är en godtycklig vektor medför att \mathbf{c} är likaså. Insättning i den svaga formen (7.6) ger

$$\int_V \mathbf{c}^T \mathbf{B}^T \boldsymbol{\sigma} \, dV = \int_S \mathbf{c}^T \mathbf{N}^T \mathbf{t} \, dS + \int_V \mathbf{c}^T \mathbf{N}^T \mathbf{b} \, dV$$

där vektorn \mathbf{c} kan elimineras eftersom den är godtyckligt vald och oberoende av läget i kroppen. Då fås:

$$\int_V \mathbf{B}^T \boldsymbol{\sigma} \, dV = \int_S \mathbf{N}^T \mathbf{t} \, dS + \int_V \mathbf{N}^T \mathbf{b} \, dV \quad (7.8)$$

Observera att de ursprungliga jämviktsekvationerna är giltiga för varje konstitutivt samband mellan spänning $\boldsymbol{\sigma}$ och töjning $\boldsymbol{\epsilon}$. Inför nu ett specifikt konstitutivt samband. Med Hookes lag fås

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$$

vilket med hjälp av approximationen i (7.7) ger

$$\boldsymbol{\sigma} = \mathbf{D}\mathbf{B}\mathbf{a} \quad (7.9)$$

Insättning av (7.9) i (7.8) ger finita elementformuleringen

$$\int_V \mathbf{B}^T \mathbf{D}\mathbf{B} \, dV \mathbf{a} = \int_S \mathbf{N}^T \mathbf{t} \, dS + \int_V \mathbf{N}^T \mathbf{b} \, dV \quad (7.10)$$

vilken förkortat omskrivs

$$\mathbf{K}\mathbf{a} = \mathbf{f}$$

med

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{D}\mathbf{B} \, dV$$

och

$$\mathbf{f} = \int_S \mathbf{N}^T \mathbf{t} \, dS + \int_V \mathbf{N}^T \mathbf{b} \, dV$$

Lösningen av jämviktsekvationerna förutsätter kännedom om randvillkoren. För varje del av kroppens yta S måste antingen ytlasten \mathbf{t} eller förskjutningen \mathbf{u} vara känd [38].


```

alpha=0.5
d=alpha*h           # CROSS SECTIONAL HEIGHT IN FRONT OF THE NOTCH
L=0.72             # LENGTH OF SIDE WITHOUT NOTCH
L_notched=0.125   # LENGTH OF SIDE WITH NOTCH
b=.006             # CROSS SECTIONAL WIDTH
beta = .5          # 0.5<BETA<1.
                   # BETA=1: ONE-SIDED NOTCH
                   # BETA=0.5: DOUBLY-EQUAL-SIDED NOTCH
e = 0.001          # LOAD EXCENTRICITY
meshSize=0.004    # MESH SIZE
meshSizeDense=meshSize # MESH SIZE IN FRACTURE PROCESS ZONE
loadApplied=20e3  # APPLIED LOAD
shearForce=0
crackRange=[meshSizeDense*i for i in range(20,23)] #PROPAGATION INTERVAL.
               # NUMBER OF ELEMENT LENGTHS OF INITIAL CRACK RANGE IS FIRST NUMBER IN "RANGE"
               # NODE RELEASE IS MADE FOR ONE NODE AT A TIME UNTIL SECOND NUMBER IN "RANGE"
GI=283           # CRITICAL STRAIN ENERGY RELEASE RATE MODE I
GII=1240         # CRITICAL STRAIN ENERGY RELEASE RATE MODE II

```

```

#####
### NODE RELEASING ###
#####
index = 0
index += 1
#####
### Create model ###
#####
m = mdb.Model(name='Notched wooden element')

#####
### Create part ###
#####
# Create a sketch for the base feature.
mySketch = m.ConstrainedSketch(name='woodenProfile',
sheetSize=1.5)
# Distinguish between load cases
if loadCase==11 or loadCase==21:
    if loadCase==11:
        d=d/2
        h=h/2
        loadApplied=loadApplied/2
    # Create the feature lines
    xyCoords = ((0,0), (L_notched,0), (L_notched,d), (-L,d), (-L,-(h-d)),
                (0,-(h-d)), (0,0))
    for i in range(len(xyCoords)-1):
        mySketch.Line(point1=xyCoords[i],
                       point2=xyCoords[i+1])
    # Create a two-dimensional planar, deformable part.
    p = m.Part(name='Part 1', dimensionality=TWO_D_PLANAR, type=DEFORMABLE_BODY)
    p.BaseShell(sketch=mySketch)
    del mySketch
    # Create a partition of the wood parallel to crack plane
    face = p.faces.findAt((0,0,0),)
    mySketch = m.ConstrainedSketch(gridSpacing=0.005, name='woodPartition',
sheetSize=1)
    mySketch.Line(point1=(0,0), point2=(-L,0))
    p.PartitionFaceBySketch(faces=face, sketch=mySketch)
    del mySketch
if loadCase==12:

```

```

# Create the feature lines
xyCoords = ((0,0), (L_notched,0), (L_notched,d), (0,d), (0,d+(1-alpha)*beta*h),
            (-L,d+(1-alpha)*beta*h), (-L,-(1-alpha)*(1-beta)*h),
            (0,-(1-alpha)*(1-beta)*h), (0,0))
for i in range(len(xyCoords)-1):
    mySketch.Line(point1=xyCoords[i],
                  point2=xyCoords[i+1])
# Create a two-dimensional planar, deformable part.
p = m.Part(name='Part 1', dimensionality=TWO_D_PLANAR, type=DEFORMABLE_BODY)
p.BaseShell(sketch=mySketch)
del mySketch
# Create a partition of the wood parallel to crack plane
face = p.faces.findAt((0,0,0),)
mySketch = m.ConstrainedSketch(gridSpacing=0.005, name='woodPartition',
                               sheetSize=1)
mySketch.Line(point1=(0,0), point2=(-L,0))
p.PartitionFaceBySketch(faces=face, sketch=mySketch)
del mySketch

# Partition the fracture path
# Partition by datum point 1
p.DatumPointByCoordinate(coords=(-crackRange[-1]-5*meshSizeDense, 0.0, 0.0))
p.PartitionEdgeByPoint(edge=p.edges.findAt(coordinates=(-meshSizeDense, 0.0, 0.0)),
                      point=p.datums[p.datums.keys()[-1]])
# Partition by datum point 2
meshTransitionFactor=40
if meshTransitionFactor*meshSizeDense+crackRange[-1]>L:
    meshTransitionFactor=(L-crackRange[-1])/meshSizeDense
p.DatumPointByCoordinate(coords=(-crackRange[-1]-meshTransitionFactor*meshSizeDense,
                                0.0, 0.0))
p.PartitionEdgeByPoint(edge=p.edges.findAt(coordinates=(-L+meshSizeDense, 0.0, 0.0)),
                      point=p.datums[p.datums.keys()[-1]])

#####
### CREATE MATERIAL ###
#####
# Create material.
myWood = m.Material(name='Wood')
# Create the elastic properties
elasticProperties = (11332000000.0, 1774000000.0, 1331000000.0, 1095000000.0, 529000000.0,
                   564000000.0, 640000000.0, 580000000.0, 30000000.0)
myWood.Elastic(type=ORTHOTROPIC, table=(elasticProperties,),
              noCompression=OFF, noTension=OFF, temperatureDependency=OFF)

#####
### CREATE SECTION ###
#####
# Create the solid section.
mySection = m.HomogeneousSolidSection(name='woodSection',
                                       material='Wood', thickness=b)
# Assign the section to the regions.
faces = p.faces.findAt(((0,meshSizeDense,0),(0,meshSizeDense,1)),((-meshSize,-
(-meshSize,-meshSizeDense,1)),)
region = Region(faces=faces)
p.SectionAssignment(offset=0.0,
                   offsetField='', offsetType=MIDDLE_SURFACE, region=region, sectionName='woodSection',
                   thicknessAssignment=FROM_SECTION)
p.MaterialOrientation(additionalRotationType=ROTATION_NONE, axis=AXIS_3, fieldName='',
                    localCsys=None, orientationType=GLOBAL, region=region, stackDirection=STACK_3)

#####

```

```

### Create instance ###
#####
a = m.rootAssembly
a.DatumCsysByDefault(CARTESIAN)
i1 = a.Instance(name='woodInstance1',
part=p, dependent=ON)

#####
### Create step ###
#####
m.StaticStep(name='woodLoad', previous='Initial',
description='Tensioning of wooden element.', timePeriod=1.0,
initialInc=1e-5, minInc=1e-9, maxInc=1.0)

#####
### CREATE MESH ###
#####
# Define the meshing region of part 1
# 1) Configure mesh controls
p.setMeshControls(technique=FREE,
elemShape=QUAD_DOMINATED, regions=p.faces)
# 2) Seed the part instance.
p.seedPart(size=meshSize)
# 3) Seed the cohesive zone
p.seedEdgeBySize(edges=p.edges.findAt((( -meshSizeDense,0,0))), size=meshSizeDense,
deviationFactor=0.1, constraint=FIXED)
p.seedEdgeByBias(biasMethod=SINGLE, end1Edges=p.edges.findAt((( -crackRange[-1]-
(meshTransitionFactor-1)*meshSizeDense,0,0))), minSize=meshSizeDense,
maxSize=meshSize, constraint=FINER)
# 4) Generate mesh
p.generateMesh()

# Regenerate assembly
a.regenerate()

#####
### Create viewport ###
#####
vp = session.Viewport(name='Notched wooden element example',
origin=(10, 10), width=200, height=100)
# Set font properties
vp.viewportAnnotationOptions.setValues(legendFont='*-verdana-medium-r-normal
--*-120*-*-p*-*-*',
titleFont='*-verdana-medium-r-normal--*-120*-*-p*-*-*',
stateFont='*-verdana-medium-r-normal--*-120*-*-p*-*-*')
# Display the meshed beam with node and element labels
vp.partDisplay.setValues(mesh=ON)
vp.partDisplay.meshOptions.setValues(meshTechnique=ON)
vp.setValues(displayedObject=p)
vp.partDisplay.meshOptions.setValues(nodeLabels=ON, elementLabels=ON)

#####
### Create Reference point ###
#####
# Create reference point
if loadCase==11:
e=0
refPointCoords=(L_notched,d/2-e,0) # Reference point coordinates
a.ReferencePoint(point=refPointCoords)
r1 = a.referencePoints

```

```

refPoints1=(r1[r1.keys()[-1]], )
region1=Region(referencePoints=refPoints1)
# Identify edge to couple with reference point
e1 = i1.edges
edges1 = e1.findAt(((L_notched, d/2, 0.0), ))
region2=Region(edges=edges1)
# Set coupling constraint
m.Coupling(name='Constraint-1', controlPoint=region1, surface=region2,
           influenceRadius=WHOLE_SURFACE, couplingType=KINEMATIC, localCsys=None,
           u1=0N, u2=0FF, ur3=0N)

#####
### Create load ###
#####
# Create point load on the reference point previously defined
mdb.models['Notched wooden element'].ConcentratedForce(name='Load-1',
               createStepName='woodLoad', region=region1, cf1=loadApplied,
               distributionType=UNIFORM, field='', localCsys=None)

# Lock the "far left" end in the horizontal direction
# 1) Find the edge by use of coordinates
endEdge1 = i1.edges.findAt((-L,meshSize,0),,)
endEdge2 = i1.edges.findAt((-L,-meshSize,0),,)
endRegion = Region(edges=endEdge1+endEdge2)
# 2) Lock in the horizontal direction
m.XsymmBC(createStepName='woodLoad', name='X-lock', region=endRegion)

# Lock the "far left" end in the vertical direction
n=2
# Number of nodes to lock in either part. One node does not suffice for convergence.
n1=i1.nodes.getByBoundingBox(xMin=-L-meshSize/2,yMin=-meshSize/2,zMin=-1,
                             xMax=-L+meshSizeDense/2,yMax=meshSize*n-meshSize/2,zMax=1)
r1=Region(nodes=n1)          # Region of nodes instance 1

# Pin the nodes
m.PinnedBC(name='Pinnedp1', createStepName='woodLoad', region=r1)

if loadCase==11:
    # Apply symmetry
    # 1) Find top edge using coordinates
    topEdge = i1.edges.findAt((-L/2, d, 0), )
    topRegion = Region(edges=topEdge)
    # 2) Create symmetry boundary condition
    m.XasymmBC(createStepName='woodLoad', name='Symmetry', region=topRegion)

#####
### Define field and history output ###
#####
m.fieldOutputRequests['F-Output-1'].setValues(variables=(
    'U', 'COORD', 'RT', 'S'))
m.historyOutputRequests['H-Output-1'].setValues(variables=(
    'ALLSE', ))

#####
### CREATE JOB FROM MODEL ###
#####
# Create an analysis job for the model and submit it.
jobName = 'NR'+str(loadCase)+'X'+str(int(meshSize*1e3))+'X'+str(int(loadApplied/1e3))+'X'
          +str(int(h*1e3))+'X'+str(int(d*1e3))+'X'+str(int((L+L_notched)*1e3))+'X'
          +str(int(L_notched*1e3))+'X'+str(int(b*1e3))+'X'+str(int(index-1))

```

```

myJob = mdb.Job(name=jobName, model='Notched wooden element',
                description='Notched wooden element under tensioning with crack propagation')
# Wait for the job to complete.
myJob.submit()
myJob.waitForCompletion()

#####
### RELEASE INITIAL NODE ###
#####
non = len(p.nodes)                                # total number of nodes
numberOfElements = len(p.elements)              # total number of elements
elementsHighlighted=p.elements.getByBoundingBox(xMin=-crackRange[-1]-2*meshSizeDense,
        yMin=-3/2*meshSize,zMin=-1,xMax=meshSizeDense,yMax=meshSizeDense/5,zMax=1)
# Edit input file
with open(jobName + '.inp', 'r') as f:          # open file read-only
    l = f.readlines()                          # store each line in a list l
# Move pointer
for i in range(len(l)):
    if '*Element, type=CPS3' in l[i] or '*Element, type=CPS4R' in l[i]:
        locus = i
        break
# Create one new node
nodes = []
nodes.append(p.nodes.getByBoundingSphere(center=(0,0,0), radius=meshSizeDense/5))
nodes.append(p.nodes.getByBoundingSphere(center=(-meshSizeDense,0,0),
        radius=meshSizeDense/5))
non += 1                                         # increment node counter
l.insert(locus, ' '+str(int(non))+', '+str(nodes[0][0].coordinates[0])+', '+
        str(nodes[0][0].coordinates[1])+'\n')
# Interesting element
elements = []
node1=0
node2=0
# Identify the element that is connected to the nodes defined in the list n o d e s
# and which lies below the partition line
for i in range(len(elementsHighlighted)):
    for j in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[j]==nodes[0][0].label-1:
            node1=1
            break
    for k in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[k]==nodes[1][0].label-1:
            node2=1
            break
    if node1==1 and node2==1:
        # Add the interesting element to the element list
        elements.append(elementsHighlighted[i])
        break
    else:
        node1=0
        node2=0

# Move pointer
if len(elements[0].connectivity)==3:
    type='CPS3'
elif len(elements[0].connectivity)==4:
    type='CPS4R'
for i in range(len(l)):
    if '*Element, type='+type in l[i]:
        locus = i

```

```

        break
# Alter element info
for i in range(locus+1,locus+len(p.elements)+2):
    listRow=l[i].split(",")
    listPart=listRow[0].split(" ")
    if str(elements[0].label)==listPart[-1]:
        if ' '+str(nodes[0][0].label)+' ' in l[i]:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+' ', ' ' +
                +str(int(non))+' ', 1)
        else:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+'\n', ' ' +
                +str(int(non))+'\n')
    break

with open(jobName + '.inp', 'w') as f:
    for i in range(len(l)):
        f.write(l[i])

#####
### Create job from input file ###
#####
myJob = mdb.JobFromInputFile(name=jobName, inputFileName='D:\\ABAQUS\\Temp\\'+jobName
    type=ANALYSIS, atTime=None, waitMinutes=0, waitHours=0, queue=None,
    memory=50, memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, userSubroutine='', scratch='',
    parallelizationMethodExplicit=DOMAIN, numDomains=1, activateLoadBalancing=False,
    multiprocessingMode=DEFAULT, numCpus=1)

# Wait for the job to complete.
myJob.submit()
myJob.waitForCompletion()

#####
### RELEASE REMAINING NODES CONSECUTIVELY ###
#####
releaseCount = 1 # ticker
crackStep=0 # crack tip pointer
while releaseCount<int(crackRange[0]/meshSizeDense):
    releaseCount += 1 # increment tickers
    if releaseCount == 1e3:
        break # safety brake
    crackStep+=meshSizeDense # increment crack tip pointer

# Edit input file
with open(jobName + '.inp', 'r') as f: # open file read-only
    l = f.readlines() # store each line in a list l
# Move pointer
for i in range(len(l)):
    if '*Element, type=CPS3' in l[i] or '*Element, type=CPS4R' in l[i]:
        locus = i
        break
# Create another node
non += 1 # increment node counter
nodes = []
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep,0,0),
l.insert(locus, ' '+str(int(non))+' '+str(nodes[0][0].coordinates[0])+
    ', '+str(nodes[0][0].coordinates[1])+'\n')
# Interesting elements
elements = []
# Element number 1 on the fractured side
nodes = []

```

```

nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep+meshSizeDense,0,0),
radius=meshSizeDense/3))
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep,0,0),
radius=meshSizeDense/3))
node1=0
node2=0
# Identify the element that is connected to the nodes defined in the list nodes
# and which lies below the partition line
for i in range(len(elementsHighlighted)):
    for j in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[j]==nodes[0][0].label-1:
            node1=1
            break
    for k in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[k]==nodes[1][0].label-1:
            node2=1
            break
    if node1==1 and node2==1:
        # Add the interesting element to the element list
        elements.append(elementsHighlighted[i])
        break
    else:
        node1=0
        node2=0

# Element number 2
nodes = []
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep,0,0),
radius=meshSizeDense/3))
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep-meshSizeDense,0,0),
radius=meshSizeDense/3))
node1=0
node2=0
# Identify the element that is connected to the nodes defined in the list nodes
# and which lies below the partition line
for i in range(len(elementsHighlighted)):
    for j in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[j]==nodes[0][0].label-1:
            node1=1
            break
    for k in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[k]==nodes[1][0].label-1:
            node2=1
            break
    if node1==1 and node2==1:
        # Add the interesting element to the element list
        elements.append(elementsHighlighted[i])
        break
    else:
        node1=0
        node2=0

# Move pointer for interesting element number 1
if len(elements[0].connectivity)==3:
    type='CPS3'
elif len(elements[0].connectivity)==4:
    type='CPS4R'
for i in range(len(l)):
    if '*Element, type='+type in l[i]:
        locus = i

```



```

        break
# Alter element properties (i.e. to which nodes it connects)
# Distinguish between the case where the node label to be replaced occurs
# at the end of the line (in the input file) and where it occurs
# in the middle or beginning
for i in range(locus+1,locus+len(p.elements)+2):
    listRow=l[i].split(",")
    listPart=listRow[0].split(" ")
    if str(elements[0].label)==listPart[-1]:
        if ' '+str(nodes[0][0].label)+' , ' in l[i]:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+' , ',' ' +
                +str(int(non))+' , ')
        else:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+'\n',' ' +
                +str(int(non))+'\n')
    break

# Move pointer for interesting element number 2
if len(elements[1].connectivity)==3:
    type='CPS3'
elif len(elements[1].connectivity)==4:
    type='CPS4R'
for i in range(len(l)):
    if '*Element, type='+type in l[i]:
        locus = i
        break
# Alter element properties (i.e. to which nodes it connects)
# Distinguish between the case where the node label to be replaced occurs
# at the end of the line (in the input file) and where it occurs
# in the middle or beginning
for i in range(locus+1,locus+len(p.elements)+2):
    listRow=l[i].split(",")
    listPart=listRow[0].split(" ")
    if str(elements[1].label)==listPart[-1]:
        if ' '+str(nodes[0][0].label)+' , ' in l[i]:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+' , ',' ' +
                +str(int(non))+' , ')
        else:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+'\n',' ' +
                +str(int(non))+'\n')
    break

with open(jobName + '.inp', 'w') as f:
    for i in range(len(l)):
        f.write(l[i])

#####
###  RELEASE NODES PRESCRIBED IN propagationInterval ###
###                               WHILE WRITING ODB                               ##
#####
G=[]
mixedModeRatio=[]
W=[]
releaseCount = 0                                     # ticker
for crackStep in crackRange:

    #####
    ### Release one node ###
    #####
    releaseCount += 1                                 # increment tickers

```

```

# Edit input file
with open(jobName + '.inp', 'r') as f: # open file read-only
    l = f.readlines() # store each line in a list l
# Move pointer
for i in range(len(l)):
    if '*Element, type=CPS3' in l[i] or '*Element, type=CPS4R' in l[i]:
        locus = i
        break
# Create another node
non += 1 # increment node counter
nodes = []
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep,0,0),
radius=meshSizeDense/3))
l.insert(locus, ' '+str(int(non))+' '+str(nodes[0][0].coordinates[0])+' ' +
+str(nodes[0][0].coordinates[1])+'\n')
# Interesting elements
elements = []
# Element number 1
nodes = []
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep+meshSizeDense,0,0),
radius=meshSizeDense/3))
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep,0,0),
radius=meshSizeDense/3))
node1=0
node2=0
# Identify the element that is connected to the nodes defined in the list n o d e s
# and which lies below the partition line
for i in range(len(elementsHighlighted)):
    for j in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[j]==nodes[0][0].label-1:
            node1=1
            break
    for k in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[k]==nodes[1][0].label-1:
            node2=1
            break
    if node1==1 and node2==1:
        # Add the interesting element to the element list
        elements.append(elementsHighlighted[i])
        break
    else:
        node1=0
        node2=0

# Move pointer for interesting element number 1
if len(elements[0].connectivity)==3:
    type='CPS3'
elif len(elements[0].connectivity)==4:
    type='CPS4R'
for i in range(len(l)):
    if '*Element, type='+type in l[i]:
        locus = i
        break
# Alter element properties (i.e. to which nodes it connects)
# Distinguish between the case where the node label to be replaced occurs
# at the end of the line (in the input file) and where it occurs
# in the middle or beginning
for i in range(locus+1,locus+len(p.elements)+2):
    listRow=l[i].split(",")

```

```

listPart=listRow[0].split(" ")
if str(elements[0].label)==listPart[-1]:
    if ' '+str(nodes[1][0].label)+' , ' in l[i]:
        l[i] = l[i].replace(' '+str(nodes[1][0].label)+' , , ' '
            +str(int(non))+' , ' )
    else:
        l[i] = l[i].replace(' '+str(nodes[1][0].label)+'\n', ' '
            +str(int(non))+'\n')
    break

# Element number 2
nodes = []
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep,0,0),
    radius=meshSizeDense/3))
nodes.append(p.nodes.getByBoundingSphere(center=(-crackStep-meshSizeDense,0,0),
    radius=meshSizeDense/3))
node1=0
node2=0
# Identify the element that is connected to the nodes defined in the list n o d e s
# and which lies below the partition line
for i in range(len(elementsHighlighted)):
    for j in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[j]==nodes[0][0].label-1:
            node1=1
            break
    for k in range(len(elementsHighlighted[i].connectivity)):
        if elementsHighlighted[i].connectivity[k]==nodes[1][0].label-1:
            node2=1
            break
    if node1==1 and node2==1:
        # Add the interesting element to the element list
        elements.append(elementsHighlighted[i])
        break
    else:
        node1=0
        node2=0

# Move pointer for interesting element number 2
if len(elements[1].connectivity)==3:
    type='CPS3'
elif len(elements[1].connectivity)==4:
    type='CPS4R'
for i in range(len(l)):
    if '*Element, type='+type in l[i]:
        locus = i
        break

# Alter element properties (i.e. to which nodes it connects)
# Distinguish between the case where the node label to be replaced occurs
# at the end of the line (in the input file) and where it occurs
# in the middle or beginning
for i in range(locus+1,locus+len(p.elements)+2):
    listRow=l[i].split(",")
    listPart=listRow[0].split(" ")
    if str(elements[1].label)==listPart[-1]:
        if ' '+str(nodes[0][0].label)+' , ' in l[i]:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+' , , ' '
                +str(int(non))+' , ' )
        else:
            l[i] = l[i].replace(' '+str(nodes[0][0].label)+'\n', ' '
                +str(int(non))+'\n')

```

```

        break
# Record info on which nodes that are closest to the crack tip
crackSurfaceNodes = [nodes[0][0].label, int(non)]

with open(jobName + '.inp', 'w') as f:
    for i in range(len(l)):
        f.write(l[i])

#####
### Create job from input file ###
#####
myJob = mdb.JobFromInputFile(name=jobName, inputFileName='D:\\ABAQUS\\Temp\\'
    +jobName+'.inp', type=ANALYSIS, atTime=None, waitMinutes=0, waitHours=0,
    queue=None, memory=50, memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, userSubroutine='',
    scratch='', parallelizationMethodExplicit=DOMAIN, numDomains=1,
    activateLoadBalancing=False, multiprocessingMode=DEFAULT, numCpus=1)
# Wait for the job to complete.
myJob.submit()
myJob.waitForCompletion()

#####
### Open the output database ###
#####
odb = openOdb(jobName+'.odb')
# Create a node set from the nodes closest to the crack tip
instanceName = 'WOODINSTANCE1'
nodeSet = odb.rootAssembly.NodeSetFromNodeLabels(name='nodeSet'+str(releaseCount),
    nodeLabels=((instanceName,crackSurfaceNodes),))
# Pick last frame
frame = odb.steps['woodLoad'].frames[-1]
dispField = frame.fieldOutputs['U']
dispTemp = dispField.getSubset(region=nodeSet)
dispSubField = dispTemp.values
deltaX = abs(dispSubField[0].data[0]-dispSubField[1].data[0])
deltaY = abs(dispSubField[0].data[1]-dispSubField[1].data[1])
mixedModeRatio.append([deltaX/(deltaX+deltaY), deltaY/(deltaX+deltaY)])
G.append(GII*deltaX/(deltaX+deltaY) + GI*deltaY/(deltaX+deltaY))
# Retrieve elastic strain energy
Utemp = odb.steps['woodLoad'].historyRegions['Assembly ASSEMBLY'].
    historyOutputs['ALLSE'].data[-1][1]
W.append(Utemp)
odb.close()
print ' '
print 'SUCCESSFULLY RELEASED '+str(releaseCount)+' NODE(S)'
print ' '

#####
### WRITE DATA TO TEXT FILE ###
#####
# correct geometry in case of symmetry
if loadCase==11:
    d=d*2
    h=h*2
    loadApplied=loadApplied*2
outFile = open(jobName + '_data.txt', 'w')
header = \
"""
*****
** Data file

```

```

** NODE RELEASING
** Load case: %s
** Mesh size: %s [m]
** Full cross sectional area: %s x %s [m2]
** Notched cross sectional area: %s x %s [m2]
** Total length: %s [m]
** Length of element on notched side: %s [m]
** Applied force: %s [kN]
** Total number of elements: %s
*****
"""
outFile.write('\n')
outFile.write(header % (loadCase,meshSize,h,b,d,b,L+L_notched,L_notched,loadApplied/1e3,
    numberOfElements))
outFile.write('Crack length [m] \t Mixed mode ratio (Mode I, Mode II) \n')
for i in range(len(G)):
    outFile.write(str(crackRange[i])+'\t'*3+str(round(mixedModeRatio[i][1],2))+
        '%'+str(round(mixedModeRatio[i][0],2))+ ' , '+'\n')
outFile.write('\n')

outFile.write('Crack length [m] \t Critical energy release rate [Nm2/m] \n')
for i in range(len(G)):
    outFile.write(str(crackRange[i])+'\t'*3+str(round(G[i],2))+'\n')
outFile.write('\n')

outFile.write('Crack length [m] \t Risk of fracture quotient \n')
for i in range(len(W)-1):
    deltaW = W[i+1]-W[i]
    ratio = deltaW/(G[i]*meshSizeDense*b)
    outFile.write(str(crackRange[i+1])+'\t'*3+str(round(ratio,2))+'\n')
outFile.write('\n')

outFile.write('Crack length [m] \t Elastic strain energy released [Nm] \n')
for i in range(len(W)):
    outFile.write(str(crackRange[i])+'\t'*3+str(round(W[i],5))+'\n')
outFile.write('\n')

outFile.write('Simplified formula (risk of fracture quotient): \n')
G=G[-1]
if loadCase==11:
    outFile.write(str(round(loadApplied**2/(4*G*b**2*9040e6)*(1/d-1/h),2))+'\n')
if loadCase==21:
    outFile.write(str(round(loadApplied**2/(2*G*b**2*h*9040e6)*(1/(d/h)-1-3*
        (1-d/h)**2)+12*e**2/(h**2*(d/h)**3)-6*e/d*(1-(d/h)),2))+'\n')

outFile.write('\n')
fshear=63e6 # mean tensile strength
outFile.write('Risk of tensile failure with strength %s MPa\n' % (round(fshear/1e6,2)))
outFile.write(str(round(loadApplied/1e3/d/b/fshear,2)))

outFile.write('\n')
outFile.write('** End of data file.')
outFile.close()

print ' '
print str(round(G,2))

# Open data file in Notepad
import webbrowser
webbrowser.open(''+jobName+'_data.txt'+''')

```



```

L=.72 # LENGTH BEHIND THE NOTCH
L_notched=.125 # LENGTH IN FRONT OF THE NOTCH
b=.006 # CROSS SECTIONAL WIDTH
alpha=.5 # PROPORTION OF BODY REMAINING IN ...
# ... FRONT OF THE NOTCH
d=h*alpha # CROSS SECTIONAL HEIGHT IN FRONT ...
# ... OF THE NOTCH
beta = .4 # 0.5<BETA<1. BETA=1: ONE-SIDED NOTCH ...
# ... BETA=0.5: DOUBLY-EQUAL-SIDED NOTCH
e = .001 # LOAD EXCENTRICITY
meshSize=0.002 # MESH SIZE
meshSizeDense=meshSize # MESH SIZE IN FRACTURE PROCESS ZONE
GI=283 # CRITICAL STRAIN ENERGY RELEASE RATE MODE I
GII=1240 # CRITICAL STRAIN ENERGY RELEASE RATE MODE II
fI=2.9*1e6 # ULTIMATE STRENGTH PERPENDICULAR TO GRAIN
fII=9.8*1e6 # ULTIMATE STRENGTH PARALLELL TO GRAIN

#####
### Create model ###
#####
m = mdb.Model(name='Notched wooden element')

#####
### Create part ###
#####
#-----
# Load case 11 or 21:
# Create two parts divided by a cohesive zone.
# Part 1 is longer and applied with a load. Part 2 is shorter
and unapplied.
# Load case 12:
# Create three parts divided by two cohesive zohes.
#-----
# PART 1:
# Create a sketch for the base feature.
mySketch = m.ConstrainedSketch(name='WoodenProfile',
sheetSize=1.5)
if loadCase==11:
    d=d/2
    h=h/2
# Create the features
if loadCase==11 or loadCase==21:
    xyCoords = ((0,0), (L_notched,0), (L_notched,d), (-L,d),
(-L,0), (0,0))
elif loadCase==12:
    xyCoords = ((0,0), (L_notched,0), (L_notched,d), (0,d), (0
,d+(1-beta)*(h-d)), (-L,d+(1-beta)*(h-d)), (-L,0), (0,0))
for i in range(len(xyCoords)-1):
    mySketch.Line(point1=xyCoords[i],
point2=xyCoords[i+1])
# Create a two-dimensional planar, deformable part.
p1 = m.Part(name='Part 1', dimensionality=TWO_D_PLANAR, type=
DEFORMABLE_BODY)
p1.BaseShell(sketch=mySketch)
del mySketch

if loadCase==11 or loadCase==21:
    # PART 2:
    # Create a sketch for the base feature.
    mySketch = m.ConstrainedSketch(name='WoodenProfile',

```



```

p1.MaterialOrientation(additionalRotationType=ROTATION_NONE,
    axis=AXIS_3, fieldName='', localCsys=None, orientationType=
    GLOBAL, region=region1, stackDirection=STACK_3)
p2.MaterialOrientation(additionalRotationType=ROTATION_NONE,
    axis=AXIS_3, fieldName='', localCsys=None, orientationType=
    GLOBAL, region=region2, stackDirection=STACK_3)

#####
### Create instance ###
#####
# Create a part instance.
a = m.rootAssembly
a.DatumCsysByDefault(CARTESIAN)
i1 = a.Instance(name='woodInstance1', part=p1, dependent=ON)
i2 = a.Instance(name='woodInstance2', part=p2, dependent=ON)

#####
### Create step ###
#####
m.StaticStep(name='woodLoad', previous='Initial', description=
    'Tensioning of wooden body.', timePeriod=1.0, initialInc=1e-5,
    minInc=1e-9, maxInc=1.0)

#####
### Create INTERACTION PROPERTIES ###
#####
# Create interaction properties
myInteraction = m.ContactProperty('IntProp-1')
myInteraction.NormalBehavior(pressureOverclosure=HARD,
    allowSeparation=ON, constraintEnforcementMethod=DEFAULT)
# Identify contact edges of the cohesive zone between parts 1
    and 2
s1 = i1.edges
side1Edges1 = s1.findAt((( -meshSize, 0.0, 0.0), ))
region1=Region(side1Edges=side1Edges1)
s2 = i2.edges
side1Edges2 = s2.findAt((( -meshSize, 0.0, 0.0), ))
region2=Region(side1Edges=side1Edges2)
# Apply interaction properties
clearance=1e-9
m.SurfaceToSurfaceContactStd(name='Int-1', createStepName=
    'woodLoad', master=region1, slave=region2, sliding=FINITE,
    thickness=ON, interactionProperty='IntProp-1', adjustMethod=
    NONE, initialClearance=OMIT, datumAxis=None, clearanceRegion=
    None)
m.interactions['Int-1'].setValues(initialClearance=clearance,
    datumAxis=None, adjustMethod=NONE, sliding=SMALL, enforcement=
    SURFACE_TO_SURFACE, thickness=ON, supplementaryContact=
    SELECTIVE, tied=OFF, bondingSet=None)
m.interactions['Int-1'].setValues(initialClearance=clearance,
    datumAxis=None, surfaceSmoothing=NONE, adjustMethod=NONE,
    sliding=SMALL, enforcement=NODE_TO_SURFACE, thickness=ON,
    supplementaryContact=SELECTIVE, smooth=0.2, bondingSet=None)

# Create reference point
if loadCase==11:
    e=0
    refPointCoords=(L_notched,d/2-e,0)# Reference point coordinates
    a.ReferencePoint(point=refPointCoords)
    r1 = a.referencePoints

```

```

refPoints1=(r1[r1.keys()[-1]], )
region1=Region(referencePoints=refPoints1)
# Identify edge to couple with reference point
e1 = i1.edges
edges1 = e1.findAt(((L_notched, d/2, 0.0), ))
region2=Region(edges=edges1)
# Set coupling constraint
m.Coupling(name='Constraint-1', controlPoint=region1, surface=
    region2, influenceRadius=WHOLE_SURFACE, couplingType=KINEMATIC,
    localCsys=None, u1=ON, u2=OFF, ur3=ON)

#####
### CREATE MESH ###
#####
# Define the meshing region of part 1
# 1) Configure mesh controls
p1.setMeshControls(technique=FREE,
    elemShape=QUAD_DOMINATED, regions=p1.faces)
# 2) Seed the part instance.
p1.seedPart(size=meshSize)
# 3) Seed the cohesive zone
p1.seedEdgeBySize(edges=p1.edges.findAt((-meshSize,0,0)),
    size=meshSizeDense, deviationFactor=0.1, constraint=FIXED)
# 4) Generate mesh
p1.generateMesh()

# Define the meshing region part 2
# 2) Configure mesh controls
p2.setMeshControls(technique=FREE,
    elemShape=QUAD_DOMINATED, regions=p2.faces)
# 3) Seed the part instance.
p2.seedPart(size=meshSize)
# 4) Seed the cohesive zone
p2.seedEdgeBySize(edges=p2.edges.findAt((-meshSize,0,0)),
    size=meshSizeDense, deviationFactor=0.1, constraint=FIXED)
# 5) Generate mesh
p2.generateMesh()

# Count number of elements
numberOfElements = len(p1.elements)+len(p2.elements)
# total number of elements

# Regenerate assembly
a.regenerate()

#####
### Create viewport ###
#####
# Create viewport
vp = session.Viewport(name='Notched wooden element example',
    origin=(10, 10), width=200, height=100)
# Set font properties
vp.viewportAnnotationOptions.setValues(legendFont=
    '-*-verdana-medium-r-normal-*-120-*-p-*-*', titleFont=
    '-*-verdana-medium-r-normal-*-120-*-p-*-*', stateFont=
    '-*-verdana-medium-r-normal-*-120-*-p-*-*')
# Display the meshed beam with node and element labels
vp.partDisplay.setValues(mesh=ON)
vp.partDisplay.meshOptions.setValues(meshTechnique=ON)
vp.setValues(displayedObject=p1)

```

```

session.viewports['Notched wooden element example'].setValues(
    displayedObject=p1)
session.viewports['Notched wooden element example'].
partDisplay.meshOptions.setValues(nodeLabels=ON, elementLabels
    =ON)

#####
### Create load ###
#####
# Create displacement control
# 1) Find the end edge by use of coordinates.
endEdge = i1.edges.findAt(((L_notched,meshSize,0),) )
endRegion = Region(side1Edges=endEdge)
# 2) Apply displacement control to the reference point
previously defined in the interactions module
m.DisplacementBC(name='Displacement', createStepName=
    'woodLoad', region=region1, u1=1, u2=UNSET, ur3=UNSET,
    amplitude=UNSET, fixed=OFF, distributionType=UNIFORM,
    fieldName='', localCsys=None)

# Lock the "far left" end in the horizontal direction
# Instance 1
# 1) Find the edge by use of coordinates
endEdge = i1.edges.findAt((-L,d/4,0),)
endRegion = Region(edges=endEdge)
# 2) Lock in the horizontal direction
m.XsymmBC(createStepName='woodLoad', name='X-lock1', region=
    endRegion)

# Instance 2
# 1) Find the edge by use of coordinates
endEdge = i2.edges.findAt((-L,-meshSize,0),)
endRegion = Region(edges=endEdge)
# 2) Lock end in the horizontal direction
m.XsymmBC(createStepName='woodLoad', name='X-lock2', region=
    endRegion)

# Lock the "far left" end in the vertical direction
n=2 # Number of nodes to lock in either
part. One node does not suffice for convergence.
n1=i1.nodes.getByBoundingBox(xMin=-L-meshSize/2,yMin=-meshSize
    /2,zMin=-1,xMax=-L+meshSizeDense/2,yMax=meshSize*n-meshSize/2,
    zMax=1) # Nodes to lock in instance 1
n2=i2.nodes.getByBoundingBox(xMin=-L-meshSize/2,yMin=-meshSize
    *(n-1)+meshSize/2,zMin=-1,xMax=-L+meshSizeDense/2,yMax=
    meshSize/2,zMax=1) # Nodes to lock in
instance 2
r1=Region(nodes=n1) # Region of nodes instance 1
r2=Region(nodes=n2) # Region of nodes instance 2
# Pin the nodes
m.PinnedBC(name='Pinnedp1', createStepName='woodLoad', region=
    r1)
m.PinnedBC(name='Pinnedp2', createStepName='woodLoad', region=
    r2)

if loadCase==11:
    # Apply symmetry
    # 1) Find top edge using coordinates
    topEdge = i1.edges.findAt((-L/2, d, 0), )
    topRegion = Region(edges=topEdge)

```

```

# 2) Create symmetry boundary condition
m.XasymmBC(createStepName='woodLoad', name='Symmetry',
            region=topRegion)

#####
### CREATE NODE SETS ###
#####
nodes1=p1.nodes.getByBoundingBox(xMin=-L-meshSizeDense/2,yMin
                                =-meshSize,zMin=-1,xMax=-L+meshSizeDense/2,yMax=d+(1-beta)*(h-
                                d)+meshSize,zMax=1)                # Set of part 1 nodes that
                                have boundary conditions
nodes2=p2.nodes.getByBoundingBox(xMin=-L-meshSizeDense/2,yMin
                                =-(h-d)-meshSize,zMin=-1,xMax=-L+meshSizeDense/2,yMax=meshSize
                                ,zMax=1)                # Set of part 2 nodes that have boundary
                                conditions
nodeSet1=p1.Set(nodes=nodes1, name='ReactionForceNodesp1')
nodeSet2=p2.Set(nodes=nodes2, name='ReactionForceNodesp2')

#####
### Define field and history output ###
#####
m.fieldOutputRequests['F-Output-1'].setValues(variables=(
    'U', 'E', 'COORD', 'RT', 'S'))

# Regenerate assembly
a.regenerate()

#####
### CREATE JOB FROM MODEL ###
#####
# Create an analysis job for the model and submit it.
jobName = 'CZM'+str(loadCase)+'X'+str(int(meshSize*1e3))+'X'+
          str(int(h*1e3))+'X'+str(int(d*1e3))+'X'+str(int((L+L_notched)*
          1e3))+'X'+str(int(L_notched*1e3))+'X'+str(int(b*1e3))
myJob = mdb.Job(name=jobName, model='Notched wooden element',
description='Wooden body with notch(es) under tensioning by
            cohesive zone modelling')
myJob.setValues(numDomains=2, activateLoadBalancing=False,
                numCpus=2)

# Wait for the job to complete.
myJob.submit()
myJob.waitForCompletion()

#####
### ASSEMBLE SPRINGS ###
#####
# Read and edit the input file
with open(jobName + '.inp', 'r') as f: # open file read-only
    l = f.readlines()                # store each line in a list l
for i in range(len(l)):              #
    if '*End Assembly' in l[i]:      #
        locus = i                    # locus points to the input line
        break                          #

# IDENTIFY NODES TO BE CONNECTED WITH SPRINGS
nodeListp1=p1.nodes.getByBoundingBox(xMin=-L-meshSizeDense/2,
                                     yMin=-meshSizeDense/1e3,zMin=-1,xMax=meshSizeDense/2,yMax=
                                     meshSizeDense/1e3,zMax=1)                # list of nodes for part 1
nodeListp2=p2.nodes.getByBoundingBox(xMin=-L-meshSizeDense/2,
                                     yMin=-meshSizeDense/1e3,zMin=-1,xMax=meshSizeDense/2,yMax=

```

```

        meshSizeDense/1e3,zMax=1)                # list of nodes for part 2
if len(nodeListp1)!=len(nodeListp2):
    print 'ERROR Number of spring pairs in the cohesive zone
do not match.'          # check equal number of nodes
# Store list of node labels and coordinates
nodeCoordsp1=[]        # list of node coordinates part 1
nodeLabelsp1=[]        # list of node labels part 1
nodeCoordsp2=[]        # list of node coordinates part 2
nodeLabelsp2=[]        # list of node labels part 2
for i in range(len(nodeListp1)):
    nodeCoordsp1.append(nodeListp1[i].coordinates[0])
    nodeLabelsp1.append(nodeListp1[i].label)
    nodeCoordsp2.append(nodeListp2[i].coordinates[0])
    nodeLabelsp2.append(nodeListp2[i].label)
# Sort list of node labels and coords with respect to the
coordinates
nodeLabelsp1Sorted=[x for y, x in sorted(zip(nodeCoordsp1,
nodeLabelsp1))]
nodeLabelsp2Sorted=[x for y, x in sorted(zip(nodeCoordsp2,
nodeLabelsp2))]
# Sort such that the node closest to the crack tip is the
first in the list
nodeLabelsp1Sorted=nodeLabelsp1Sorted[::-1]
nodeLabelsp2Sorted=nodeLabelsp2Sorted[::-1]
nodeCoordsp1=sorted(nodeCoordsp1)[::-1]
nodeCoordsp2=sorted(nodeCoordsp2)[::-1]

# INSERT SPRING ELEMENTS INTO INPUT FILE
kappa0=1e-3          # fraction of the ultimate displacement for
which linear behaviour is satisfied
deltaUltimateModeI=2*GI/fI/(1-kappa0)# ultimate displacement
of Mode I spring
deltaUltimateModeII=2*GII/fII/(1-kappa0)    # ultimate
displacement of Mode II spring
nos=len(nodeLabelsp1Sorted)    # Number of springs on either
side of cohesive zone
tnos=2*nos          # Total number of springs on all sides of all
cohesive zones

# Mode I springs
l.insert(locus,'0., 0.\n'+str(fI*b*meshSizeDense)+'', '+str(
deltaUltimateModeI*kappa0)+'\n'+'+0, '+str(deltaUltimateModeI)+
'\n')
l.insert(locus,'2, 2\n')
l.insert(locus,'*Spring, elset=SpringsModeI, nonlinear\n')
counter = -1
for i in range(nos):
    counter += 1
    l.insert(locus, str(tnos-counter)+'', woodInstance1.'+str(
nodeLabelsp1Sorted[i])+', woodInstance2.'+str(
nodeLabelsp2Sorted[i])+'\n')
l.insert(locus,'*Element, type=Spring2, elset=SpringsModeI\n')

# Mode II springs
l.insert(locus,'0., 0.\n'+str(fII*b*meshSizeDense)+'', '+str(
deltaUltimateModeII*kappa0)+'\n'+'+0, '+str(deltaUltimateModeII
)+'\n')
l.insert(locus,'1, 1\n')
l.insert(locus,'*Spring, elset=SpringsModeII, nonlinear\n')
for i in range(nos):

```

```

        counter += 1
        l.insert(locus, str(tnos-counter)+'', woodInstance1.'+str(
            nodeLabelsp1Sorted[i])+', woodInstance2.'+str(
            nodeLabelsp2Sorted[i])+'\n')
l.insert(locus, '*Element, type=Spring2, elset=SpringsModeII)

# WRITE TO INPUT FILE
with open(jobName + '.inp', 'w') as f:
    for i in range(len(l)):
        f.write(l[i]) # write each line to the input file

#####
### Create job from input file ###
#####
myJob = mdb.JobFromInputFile(name=jobName, inputFileName=
    'C:\SIMULIA\Temp\'+jobName+'.inp', type=ANALYSIS, atTime=
    None, waitMinutes=0, waitHours=0, queue=None, memory=50,
    memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE,
    userSubroutine='', scratch='', parallelizationMethodExplicit=
    DOMAIN, numDomains=1, activateLoadBalancing=False,
    multiprocessingMode=DEFAULT, numCpus=1)

# Wait for the job to complete.
myJob.submit()
myJob.waitForCompletion()

#####
### Open the output database ###
#####
# # Open the output database.
odb = openOdb(jobName+'.odb')
# Define variables
springModeIIForce = [] # list of Mode II spring forces
springModeIForce = [] # list of Mode I spring forces
springModeIIDisplacement = [] # list of Mode II spring
displacements
springModeIDisplacement = [] # list of Mode I spring
displacements
frameIndex=1 # pointer to the current frame

# LOOP OVER ALL FRAMES UNTIL SPRING FAILURE IS REACHED, OR
ELSE BREAK
failure=0 # failure counter
frameIndex=-1 # frame counter
failureFactorWhileLoop=[] # failure factor per frame
while frameIndex<len(odb.steps['woodLoad'].frames):
    frameIndex+=1
    frame = odb.steps['woodLoad'].frames[frameIndex]
    # set frame
    springValues=frame.fieldOutputs['S'].getSubset(elementType
    ='SPRING2').values
    springDisplacements=frame.fieldOutputs['E'].getSubset(
    elementType='SPRING2').values
    # Separate information about forces and displacements for
    Mode I and II into lists
    for i in range(len(springValues)/2):
        springModeIIForce.append(springValues[i].data[0])
        # Mode II forces
        springModeIForce.append(springValues[i+len(
        springValues)/2].data[0]) # Mode I forces

```

```

springModeIIDisplacement.append(springDisplacements[i
].data[0]) # Mode II displ.
springModeIDisplacement.append(springDisplacements[i+
len(springDisplacements)/2].data[0]) # Mode I displ.
# Flip the vectors so that the initial values correspond
to springs closest to the crack tip
springModeIIForce[::-1]
springModeIForce[::-1]
springModeIIDisplacement[::-1]
springModeIDisplacement[::-1]

# LOOP OVER ALL SPRING PAIRS AND EVALUATE FAILURE
CONDITION ACCORDING TO QUADRATIC POWER LAW
delta0ModeI=deltaUltimateModeI*kappa0 # greatest
displacement of elastic part of spring behaviour Mode I
delta0ModeII=deltaUltimateModeII*kappa0 # greatest
displacement of elastic part of spring behaviour Mode II
GIEff=0
GIIEff=0
failureFactorForLoop=[]
for pairIndex in range(len(springModeIForce)):
# IS MODE I SPRING IN THE ELASTIC PART?
if springModeIDisplacement[pairIndex] <= delta0ModeI:
GIEff=0
# IS MODE I SPRING IS IN THE NON-LINEAR PART?
elif springModeIDisplacement[pairIndex] > delta0ModeI:
# Calculate an effective energy release rate by
integration of the force-displacement graph
GIEff=fI*b*meshSizeDense*(deltaUltimateModeI-
delta0ModeI)/2 - springModeIForce[pairIndex]*(
deltaUltimateModeI-springModeIDisplacement[
pairIndex])/2
# IS MODE II SPRING IN THE ELASTIC PART?
if springModeIIDisplacement[pairIndex] <= delta0ModeII:
GIIEff=0
# IS MODE II SPRING IS IN THE NON-LINEAR PART?
elif springModeIIDisplacement[pairIndex] >
delta0ModeII:
# Calculate an effective energy release rate by
integration of the force-displacement graph
GIIEff=fII*b*meshSizeDense*(deltaUltimateModeII-
delta0ModeII)/2 - springModeIIForce[pairIndex]*(
deltaUltimateModeII-springModeIIDisplacement[
pairIndex])/2

# Evaluate interaction by a quadratic power law
failureFactorForLoop.append((GIEff/(GI*b*meshSizeDense
)**1 + (GIIEff/(GII*b*meshSizeDense))**1)
if failureFactorForLoop[-1] >= 1:
print 'Failure reached at spring pair number: '+
str(pairIndex+1)+' from the crack tip\n Frame
number: '+str(frameIndex)
riskModeI=springModeIDisplacement[pairIndex]/
deltaUltimateModeI
riskModeII=springModeIIDisplacement[pairIndex]/
deltaUltimateModeII
if riskModeI/riskModeII > 1:
print 'Mode I dominates mixed mode'
elif riskModeI/riskModeII < 1:
print 'Mode II dominates mixed mode'

```



```

        failure=1
        break

    if failure==1:
        # FIND THE ULTIMATE LOAD BY INVESTIGATION OF THE
        # REACTION FORCES
        reactionForcesp1=frame.fieldOutputs['RT'].getSubset(
            region=odb.rootAssembly.instances['WOODINSTANCE1'].
            nodeSets['REACTIONFORCENODESP1']).values      # list of
            reaction forces
        reactionForcesp2=frame.fieldOutputs['RT'].getSubset(
            region=odb.rootAssembly.instances['WOODINSTANCE2'].
            nodeSets['REACTIONFORCENODESP2']).values      # list of
            reaction forces
        reactionForce=0 # define the ultimate reaction force
        for i in range(len(reactionForcesp1)): # loop over
            all reaction forces
            reactionForce+=reactionForcesp1[i].data[0]
        for i in range(len(reactionForcesp2)): # loop over
            all reaction forces
            reactionForce+=reactionForcesp2[i].data[0]
        reactionForce=abs(reactionForce)                  # take the
            absolute value of the reaction
        print 'Reaction force:'+str(reactionForce/1e3)+'kN'
        print 'Tension:'+str(reactionForce/b/d/1e6)+'MPa'
        # set the critical release energy rate (mixed mode)
        # for the analytical solution
        G = (((GIEff/b/meshSizeDense)/GI)**1*GI + (((GIIEff/b/
            meshSizeDense)/GII)**1*GII)) / 1*( (((GIEff/b/
            meshSizeDense)/GI)**1 + ((GIIEff/b/meshSizeDense)/GII
            )**1) )
        break
    elif failure==0:
        failureFactorWhileLoop.append(max(failureFactorForLoop
            ))
        if frameIndex==len(odb.steps['woodLoad'].frames)-1:
            print 'frame index limit reached'
            break
odb.close()      # close output database

#####
### WRITE DATA TO TEXT FILE ###
#####
# correct geometric parameters in case of symmetry
if loadCase==11:
    d=d*2
    h=h*2
    reactionForce=2*reactionForce

outFile = open(jobName + '_data.txt', 'w')
header = \
"""
*****
** Data file.
** Cohesive zone model
** Load case: %s
** Load case counter: %s
**
**
** Approx. mesh size: %s [m]

```

```

** Full cross sectional area: %s x %s [m2]
** Cross sectional area on notched side: %s x %s [m2]
** Total length: %s [m]
** Length of notch: %s [m]
** Applied force: %s [kN]
** Equivalent tension: %s [MPa]
** Total number of elements: %s
*****
"""

outfile.write(header % (loadCase,labLoadCaseCounter,meshSize,h
    ,b,d,b,L+L_notched,L_notched,round(reactionForce/1e3,4),round(
    reactionForce/b/h/1e6,3),numberOfElements))
outfile.write('\n')

outfile.write('Gc =' +str(G)+'\n')
outfile.write('\n')

outfile.write('Tension/strength according to simplified
formula: \n')
if loadCase==11:
    outfile.write(str((reactionForce)**2/(4*G*b**2*9040e6)*(1/
        d-1/h))+'\n')
if loadCase==12:
    outfile.write(str((reactionForce)**2/(2*G*b**2*h*9040e6
        )*(-1+1/(alpha+beta*(h-d)/h)+12*(-e+beta*(h-d)/2)**2/(h**2
        *(alpha+beta*(h-d)/h)**3)-12*(-e+(2*beta*(h-d)/h+alpha-1)*
        h/2)**2/h**2))+'\n')
if loadCase==21:
    e=-e #
    outfile.write(str((reactionForce)**2/(2*G*b**2*h*9040e6)*(
        1/(d/h)-1-3*(1-d/h)**2)+12*e**2/(h**2*(d/h)**3)-6*e/d*(1-(
        d/h)))+'\n')
outfile.write('\n')
fshear=63e6 # mean shear strength
outfile.write('Risk of shear failure with strength %s MPa\n' %
    (fshear/1e6))
outfile.write(str(reactionForce/1e3/d/b/fshear))

outfile.write('\n')
outfile.write('** End of data file.')
outfile.close()

# Open data file in Notepad
import webbrowser
webbrowser.open('"+jobName+'_data.txt'+"'")

```

Bilaga 5

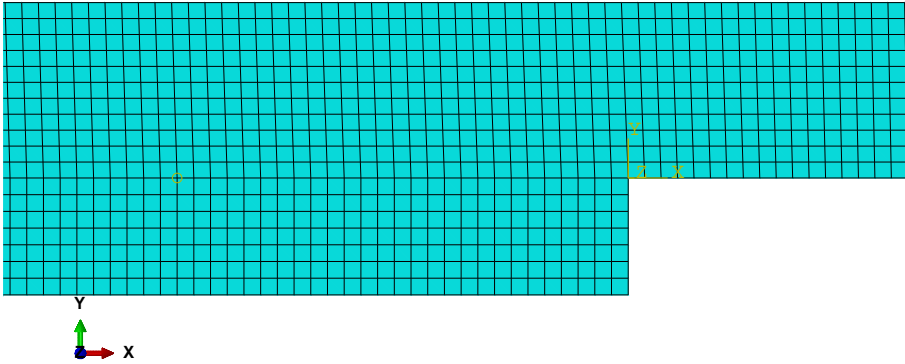


Illustration genererad i ABAQUS av finita elementnät som använts vid simulering med noduppsläppning. Illustrationen omfattar inte hela elementnätet utan en del närmast urtaget.

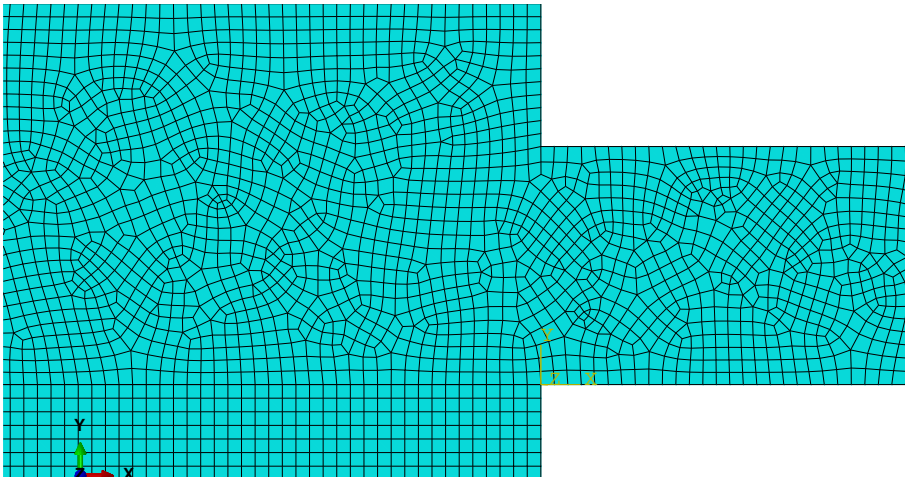


Illustration genererad i ABAQUS av finita elementnätet som använts vid simulering med kohesiv zon. Illustrationen omfattar inte hela elementnätet utan en del närmast urtagen.