

CHALMERS



Optimal design av strukturer

Utvärdering av problem och lösningsförslag inom topologioptimering

Kandidatarbete i Tillämpad mekanik

ALBIN BRANTIN
ELIAS BÖRJESSON
JAKOB EKERMO
FREDRIK EKRE

Institutionen för Tillämpad mekanik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2014
Kandidatarbete 2014:01

KANDIDATARBETE I TILLÄMPAD MEKANIK

Optimal design av strukturer

Utvärdering av problem och lösningsförslag inom topologioptimering

ALBIN BRANTIN
ELIAS BÖRJESSON
JAKOB EKERMO
FREDRIK EKRE

Institutionen för Tillämpad mekanik
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2014

Optimal design av strukturer
Utvärdering av problem och lösningsförslag inom topologioptimering
ALBIN BRANTIN
ELIAS BÖRJESSON
JAKOB EKERMO
FREDRIK EKRE

© ALBIN BRANTIN , ELIAS BÖRJESSON , JAKOB EKERMO , FREDRIK EKRE , 2014

Kandidatarbete 2014:01
ISSN 1654-4676
Institutionen för Tillämpad mekanik
Chalmers tekniska högskola
SE-412 96 Göteborg
Sverige
Telefon: +46 (0)31-772 1000

Omslag:
Två optimerade fritt upplagda balkar med rektangulära 4-nodselement (överst) och rektangulära 9-nodselement (nederst)

Chalmers Reproservice
Göteborg, Sverige 2014

Optimal design av strukturer
Utvärdering av problem och lösningsförslag inom topologioptimering
Kandidatarbete i Tillämpad mekanik
ALBIN BRANTIN
ELIAS BÖRJESSON
JAKOB EKERMO
FREDRIK EKRE
Institutionen för Tillämpad mekanik
Chalmers tekniska högskola

SAMMANFATTNING

Topologioptimering är ett industriellt verktyg vars användning har ökat kraftigt i takt med att den datoriserade beräkningskapaciteten ökat. Grundtanken är att utifrån en given domän och lastfall iterera fram en optimal struktur med hjälp av finita elementmetoden (FEM) samt optimeringsalgoritmer. Optimeringen sker med hänsyn till villkor som designomän, laster och randvillkor. För att kunna genomföra optimeringen kopplas strukturen till en målfunktion. Denna målfunktion är ofta i form av en vald strukturens respons, till exempel spänning eller elastisk töjningsenergi. Det övergripande målet med topologioptimering är att finna en konceptdesign som sedan kan vidareutvecklas. Detta kan framförallt spara mycket tid och material. Trots den positiva trenden är verktyget fortfarande i utvecklingsstadiet och associeras med ett par karaktäristiska problem som yttrar sig i form av att genererade strukturer ej är produktionsmässigt realiserbara. Tre vanliga problem som studerats i detta projekt är *checkerboarding*, *densitetsvariationer* och *nätberoende*.

Syftet med projektet var att undersöka de vanliga problem som uppkommer vid topologioptimering. Potentiella lösningar till problemen skulle studeras och testas för att undersöka vilka av dem som är mest effektiva. För att kunna göra jämförelsen utformades ett antal analysmetoder. Dessutom skulle ett användarvänligt program för topologioptimering utformas för att kunna hantera godtyckliga lastfall och domäner i två dimensioner.

För att minimera eller helt avhjälpa de karaktäristiska problem som ofta uppkommer vid topologioptimering har olika lösningsmetoder studerats. De metoder som implementerats och utvärderats i projektet är olika elementtyper samt olika filtreringsalgoritmer. Bäst resultat har uppnåtts då hexagonelement har använts, då dessa förhindrar uppkomsten av samtliga undersökta problem. De olika filtreringsalgoritmerna har också visat på goda resultat, framförallt när det gäller att generera enkelt tillverkningsbara strukturer.

Resultaten har legat till grund för ett antal slutsatser. Den första slutsatsen är att valet av elementtyp starkt påverkar slutresultatet för framgenererade strukturer i de fall filter inte använts. Dessutom dras slutsatsen att filter kraftigt påverkar slutresultatet och att det därför är viktigt att ha kontroll på hur mycket lösningen modifieras.

I takt med att tekniken går framåt spås topologioptimering få en mer framträdande roll i framtida produktutvecklingsprocesser. I dagsläget har topologioptimering fortfarande stor utvecklingspotential vad gäller en breddning av dess användningsområden. Det kommer i framtiden bli allt viktigare att vara resurseffektiv och detta verktyg tros kunna bidra till en minskad materialåtgång och minskad påverkan på miljön.

Nyckelord: Topologioptimering, Checkerboarding, Densitetsvariation, Nätberoende

ABSTRACT

Topology optimization is an industrial tool whose use has increased rapidly along with the evolution of the computational capacity. The basic idea is to find an optimal structure with respect to a chosen objective function using the finite element method (FEM) and specific optimization algorithms. This is done with respect to restrictions such as a design domain and boundary conditions. The objective function is often expressed in the form of a physical quantity, such as stress or elastic strain energy. The overall goal of topology optimization is to find a concept design for further development. This method can save time as well as resources. Despite the positive trend, the tool is still in need of development, and is often associated with a number of characteristic problems that appear in the generated structures.

The aim of the project was to examine the common issues that arise in topology optimization. Solutions to the problems were to be studied and implemented, and then analyzed in order to find the most effective one. To make the comparison possible, a number of methods for analysis have been developed. Furthermore, a user friendly topology optimization program was to be designed to handle arbitrary loading cases and domains in two dimensions.

Two types of solution methods for the characteristic problems have been studied; the use of different element types in the FE-analysis, and different filtering algorithms that manipulate the solution in a desirable way. The best results have been achieved using hexagon elements, as these prevent all of the investigated problems. The different filtering algorithms have also yielded good results, especially with respect to manufacturability of the structures.

The results have been the basis for a number of conclusions. The first conclusion is that the choice of element type strongly affects the resulting structures. Furthermore, the use of the filtering algorithms has been proved to heavily affect the final result, and one should be aware of to what extent the filter manipulates the solution.

The use of topology optimization in future product development processes is predicted to increase, as the computational capacity and the technical knowledge is steadily improving. Assessment of the tool leads to the conclusions that it has reached a high level of maturity but still has large potential to be further developed. Future applications will put an increasingly large emphasis on resource efficiency and this tool is predicted to enable a reduced material consumption.

FÖRORD

Detta arbete är en del av kandidatexamen för en grupp bestående av fyra studenter inskrivna på civilingenjörsprogrammen Maskinteknik samt Väg- och Vattenbyggnad på Chalmers tekniska högskola. Kandidatarbetet omfattar 15 högskolepoäng vilket motsvarar en termins halvtidsstudier. Valet att göra ett arbete inom topologioptimering har baserats på det gemensamma intresset för finita elementmetoden samt dess tillämpningar. Det har legat i studenternas, men även i handledarens intresse att lära sig mer om topologioptimering. Vi vill rikta ett stort tack till vår handledare Rebecka Brommesson för den vägledning och respons vi fått under projektets gång.

INNEHÅLL

Sammanfattning	i
Abstract	ii
Förord	iii
Innehåll	v
1 INLEDNING	1
1.1 Syfte och frågeställningar	2
1.2 Avgränsningar	2
2 TEORI	3
2.1 Finita elementformulering av jämviktsekvationen	4
2.2 Matematisk formulering av topologioptimeringsproblemet	4
2.3 Vanliga problem vid topologioptimering	6
2.3.1 Checkerboarding	6
2.3.2 Densitetsvariation	8
2.3.3 Nätberoende	9
2.4 Olika elementtyper	9
2.4.1 Rektangulära element med fyra noder	10
2.4.2 Rektangulära element med fyra noder och rotationsfrihetsgrader	10
2.4.3 Rektangulära element med nio noder	11
2.4.4 Hexagonala element med sex noder	11
2.5 Topologioptimeringsprogrammet top88	12
3 METOD	13
3.1 Referensfall	13
3.2 Implementerade lösningsförslag	13
3.3 Analysmetoder	14
3.4 Utformning av grafiskt gränssnitt	15
4 RESULTAT	16
4.1 Olika elementtyper	16
4.1.1 Rektangulära element med fyra noder	16
4.1.2 Rektangulära element med fyra noder och rotationsfrihetsgrader	16
4.1.3 Rektangulära element med nio noder	17
4.1.4 Förskjutna Rektangulära element med nio noder	18
4.1.5 Hexagonala element med sex noder	18
4.2 Filter	19
4.3 Jämförelse mellan lösningsmetoder	20
4.4 Grafiskt gränssnitt för topologioptimering	21
5 DISKUSSION OCH SLUTSATSER	22
Referenser	23
Bilaga A Ytterligare strukturer med filter och varierande R_{min}	
Bilaga B Sammanställda tabeller för samtliga referensfall	

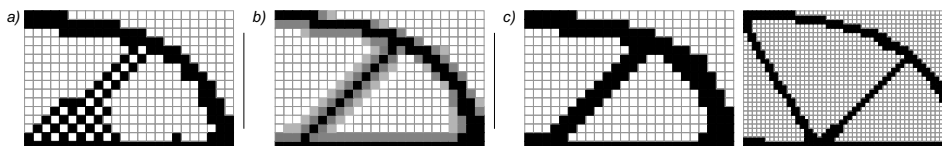
1 INLEDNING

Topologioptimering är ett ämne som har fått en allt mer central roll vid design av lastbärande strukturer i takt med att utvecklingen av datorkapacitet har gått framåt. Ämnet klassas som en undergren till det mer generella området strukturoptimering, som brukar delas in i tre klasser enligt nedan [4]:

- *Storleksoptimering* innebär att geometrin för strukturen är definierad samt att optimeringen sker med någon slags storlek som variabel, till exempel tjockleken av en platta eller tvärsnittsarean av en balk.
- *Formoptimering* innebär att topologin (antal hål, balkar etc) redan är känd. Optimeringen sker här med någon slags geometri som variabel, till exempel en kroppscontur.
- *Topologioptimering* är den mest generella grenen och kännetecknas av att det enda som är känt är designdomänen, dvs den rymd som lösningen tillåts befinna sig i. Både topologin, storleken och formen är obekant.

Inom mekaniken innebär topologioptimering att en optimal struktur med avseende på valda kriterier itereras fram utifrån en given domän och givna parametrar. Optimeringen kan ske med avseende på många olika målfunktioner som till exempel spänning eller utböjning. Ett vanligt fall är att styvheten för strukturen maximeras för en given mängd material. För att kunna genomföra optimeringen används finita elementmetoden (FEM) och ändamålsenliga algoritmer. Fördelarna med användandet av topologioptimering är många. Vid implementering tidigt i designprocessen kan projekteringstiden förkortas markant. Med hjälp av optimeringen fås en konceptdesign som kan agera utgångspunkt för vidareutveckling. Den framtagna designen behöver sedan endast finjusteras och kontrolleras med avseende på mekaniska egenskaper. Om optimering sker med avseende på styvhet krävs ofta att sekundära storheter som till exempel spänning kontrolleras för att säkerställa hållfasthetsmässig prestanda. Med denna metodik undviks det långa, manuella samt iterativa tillvägagångssätt som annars krävs för att designa en produkt. En annan fördel är att mängden material kan minskas vilket möjliggör en högre resurseffektivitet. En minskad materialåtgång är önskvärd för att reducera kostnader och minska miljöpåverkan. Det är därför naturligt att industrin är intresserad av tekniken och vill se en stadig utveckling på området.

Användandet av topologioptimering är inte helt problemfritt eftersom ett antal vanliga komplikationer förekommer som påverkar lösningen [4]. Tre vanliga problem är *checkerboarding*, *densitetsvariationer* och *nätberoende*, se Figur 1.1. Det förstnämnda innebär att den genererade strukturen innehåller ett regelbundet schackmönster av material. Problemet med densitetsvariationer innebär att materialet i strukturen inte är homogent. Lösningen av optimeringsproblemet kan också vara beroende av vald nätupplösning för FE-diskretiseringen vilket gör att olika strukturer fås för olika nätupplösningar. Det finns olika metoder för att reducera dessa problems inverkan på lösningen. Metoderna kan bestå i att använda andra elementtyper i FE-analysen, filtreringsalgoritmer samt olika typer av efterkonstruktioner.



Figur 1.1: De tre vanliga problemen; a) checkerboarding, b) densitetsvariation och c) nätberoende.

Projektet utgick från en 88-raders MATLAB-kod [2] för topologioptimering, i fortsättningen kallad *top88*, skriven vid *Department of Solid Mechanics, Technical University of Denmark*. Koden är ämnad för utbildningssyfte och fungerar som en introduktion till topologioptimering. Som komplement till koden finns en artikel som beskriver programmet med avseende på uppbyggnad, aktuell teori samt ger ett antal förslag till metoder som kan implementeras för att lösa ovannämnda problem [2].

1.1 Syfte och frågeställningar

Syftet med projektet var att undersöka de vanliga problemen som uppkommer vid topologioptimering. Potentiella lösningar till problemen skulle studeras och implementeras i top88. Lösningarna skulle testas och utvärderas för att undersöka vilka av dem som är mest effektiva. Vidare skulle den nämnda MATLAB-koden göras mer generell och användarvänlig. Generaliseringen har här den specifika innebörden att programmet skulle kunna hantera godtyckliga tvådimensionella och stationära lastfall. Programmet skulle även kunna visualisera resulterande spänningar och utböjningar i de framtagna strukturerna.

För att ytterligare precisera syftet bröts det ner i ett antal frågeställningar vilka redovisas nedan:

- Hur påverkas uppkomsten av problemen i lösningen av vald elementtyp?
- Hur påverkas uppkomsten av problemen i lösningen av filter?
- Hur mycket skiljer beräkningstiden för de olika lösningsförslagen?

1.2 Avgränsningar

Nedan presenteras de avgränsningar som preciserade projektets ramar:

- Optimering kommer endast att ske med komplians (inverterad styvhet) som målfunktion.
- Endast stationära lastfall kommer att studeras.
- Endast tvådimensionella lastfall kommer att studeras.
- Endast linjärelastiskt material och plan spänning kommer att studeras.
- Ingen hänsyn kommer tas till andra faktorer än de mekaniska, till exempel kommer produktionsmässiga och ekonomiska kriterier i designprocessen bortses från.

Det som ligger utanför dessa punkter kan ofta vara viktigt för industriella applikationer, dock avgränsas projektet på grund av begränsade tids- och resursramar.

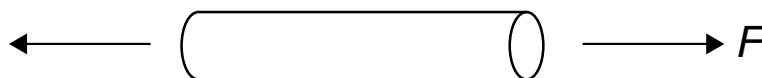
2 TEORI

Optimering innebär att utifrån ett givet problem samt givna parametrar och villkor hitta den lösning som är bäst med avseende på ett eller flera kriterier. En mekanisk struktur har som huvuduppgift att bära laster och optimering inom strukturanalys har därmed den specifika innebörden att hitta den bäst lämpade strukturen för ett givet lastfall [4]. För att konkretisera problemet relateras strukturens karaktäristik till en eller flera målfunktioner (*eng. objective functions*), vilka är formulerade så att en eller flera storheter skall minimeras eller maximeras. Den optimala strukturen är följaktligen den som maximerar eller minimerar den givna målfunktionen. Exempel på vanligt använda storheter är:

- Massa
- Spänning
- Utböjning
- Komplians (elastisk töjningsenergi)
- Produktionskostnad

Valet av målfunktion sker i regel som en direkt följd av vilken typ av mekaniskt problem som studeras. Exempelvis efterfrågar en balk som utsätts för böjning andra mekaniska parametrar än om den utsätts för utmattning. Det går även att ha flera målfunktioner för sitt problem [4].

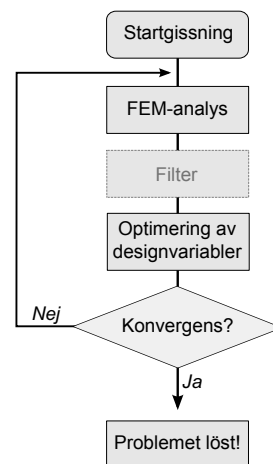
Det följer rent intuitivt att vissa begränsningar måste introduceras för att ett optimeringsproblem skall ha en entydig lösning. För att illustrera detta är ett möjligt fall en stång med godtyckligt tvärsnitt som utsätts för en axiell kraft, se Figur 2.1.



Figur 2.1: Stång utsatt för axiell dragkraft.

Om målfunktionen för denna struktur är att minimera dragspänningar i stångens tvärsnitt skulle en lösning på optimeringsproblemet ej kunna erhållas eftersom tvärsnittet kan göras oändligt stort och därmed ge upphov till infinitesimala spänningar. Många olika typer av begränsningar kan införas för att undvika detta problem. Exempelvis kan tillgängligt material eller geometri begränsas eller en designdomän införas. Med det sistnämnda menas en rymd i vilken strukturen tillåts befinna sig i.

Topologioptimering är en iterativ process som åskådliggörs av den schematiska skissen i Figur 2.2. Första steget är att ställa upp en startgissning som sedan analyseras med hjälp av FEM. Därefter appliceras eventuella filter som påverkar lösningen. Vidare optimeras designvariablerna med avseende på målfunktion med hänsyn till bivillkor. Sista steget kontrollerar om lösningen konvergerat. Om tillräcklig noggrannhet ej är uppnådd görs en ny iteration med de uppdaterade designvariablerna; annars anses problemet vara löst [7].



Figur 2.2: Schematisk beskrivning av den iterativa processen i topologioptimering.

2.1 Finita elementformulering av jämviktsekvationen

Den differentialekvation som är av intresse (vid optimering av komplians) är elasticitetsekvationen som beskriver jämviktsambanden för en kontinuerlig kropp, och ges enligt:

$$-\tilde{\nabla}^T \boldsymbol{\sigma} = \mathbf{b} \quad (2.1)$$

där $\tilde{\nabla}$ är nablaoperatoren på Voigt-form, $\boldsymbol{\sigma}$ är spänningsvektorn och \mathbf{b} är en volymslast. Denna kontinuerliga ekvation diskretiseras med hjälp av finita elementmetoden vilket ger ett ekvationssystem enligt:

$$\mathbf{K}\mathbf{U} = \mathbf{F} \quad (2.2)$$

där \mathbf{K} är den globala styvhetsmatrisen, \mathbf{U} är en vektor med nodvariabler och \mathbf{F} är lastvektorn. Styvhetsmatrisen \mathbf{K} fås genom assemblering av elementstyvhetsmatriserna \mathbf{K}^e , vilka ges av:

$$\mathbf{K}^e = \int_{A^e} \mathbf{B}^{eT} \mathbf{D} \mathbf{B}^e dA^e \quad (2.3)$$

där \mathbf{D} är den konstitutiva matrisen och \mathbf{B}^e är derivator av elementets basfunktioner. För en komplett genomgång av den bakomliggande teorin hänvisas läsaren till *An Introduction to the Finite Element Method* [9].

2.2 Matematisk formulering av topologioptimeringsproblemet

För att konkretisera optimeringsproblemet införs en matematisk formulering där den generella formen ser ut enligt:

$$\begin{cases} \min_{x_i, y_i} : & f(\mathbf{x}, \mathbf{y}); \\ x_{i,\min} \leq x_i \leq x_{i,\max} \\ y_{j,\min} \leq y_j \leq y_{j,\max} \end{cases} \quad (2.4)$$

där $f(\mathbf{x}, \mathbf{y})$ är målfunktionen som skall minimeras, \mathbf{x} är en vektor med designvariabler, det vill säga parametrar som beskriver geometrin, och \mathbf{y} är en vektor med tillståndsvariabler vilka representerar strukturens respons i form av spänningar, förskjutningar etc. Begränsningar införs på variablerna i form av tillåtna intervall. Innebörden av dessa begränsningar är ofta villkor som är karaktäristiska för mekanisk analys, till exempel jämviktsvillkor, konstitutiva samband samt kompatibilitetsvillkor.

Den optimering som är aktuell för projektet sker med avseende på styvhet. Den inverterade styvheten kallas komplians och kan tolkas som den elastiska töjningsenergin. Styvhetsoptimering innebär därmed att hitta den styvaste strukturen genom att minimera dess komplians. Detta optimeringsproblem formuleras enligt följande ekvation [2]:

$$\begin{cases} \min_{\mathbf{x}} : c(\mathbf{x}) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N E_e(x_e) \mathbf{u}_e^T \mathbf{K}_e^0 \mathbf{u}_e \\ \text{s.a:} \begin{cases} \frac{V(\mathbf{x})}{V_0} = f \\ \mathbf{K} \mathbf{U} = \mathbf{F} \\ 0 \leq \mathbf{x} \leq 1 \end{cases} \end{cases} \quad (2.5)$$

där c är kompliansen, \mathbf{u}_e är en elementförskjutningsvektor, \mathbf{K}_e^0 är en elementstyvhetsmatris för ett element med normerad elasticitetsmodul, $V(\mathbf{x})$ är materialets volym, V_0 är domänens volym, f är volymfraktionen, dvs hur stor del av domänen som ska fyllas med material och \mathbf{x} är en vektor med designvariabler som varierar mellan 0 och 1, i detta fall densiteten för elementen. För att lösa problemet minimeras c med hänsyn till bivillkor på jämvikt, given volymfraktion och tillåtet intervall för \mathbf{x} .

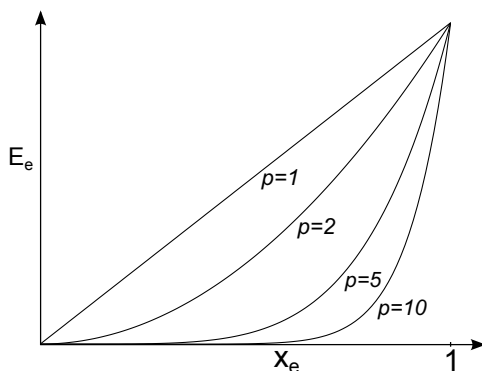
För att kunna lösa ekvation (2.5) måste en ansats för elementstyvhetsmatrisen göras. Ett populärt alternativ som har visat sig effektivt är den så kallade SIMP¹-metoden [4]. Metoden går ut på att relatera ett elements densitet, x_e , till dess styvhet, E_e , enligt:

$$E_e(x_e) = E_{\min} + x_e^p(E_0 - E_{\min}) \quad (2.6)$$

där E_0 är materialets E-modul och E_{\min} en liten, artificiell styvhet för att förhindra att styvhetsmatrisen blir singular och p är en penaliseringsfaktor. Elementstyvhetsmatrisen \mathbf{K}_e fås nu som produkten av den normerade styvhetsmatrisen \mathbf{K}_e^0 och E-modulen E_e enligt:

$$\mathbf{K}_e = \mathbf{K}_e^0(E_{\min} + x_e^p(E_0 - E_{\min})) \quad (2.7)$$

Eftersom densiteten varierar mellan 0 och 1 fås en lägre styvhet för element med intermediära densiteter på grund av faktorn p , vilket Figur 2.3 illustrerar. Det innebär att det blir mindre materialeffektivt att ta fram strukturer innehållande element med intermediära densiteter. SIMP bidrar alltså till att ge lösningar utan densitetsvariationer. Nackdelar med metoden är dock att lösningen blir beroende av penaliseringsfaktorn p och ger upphov till checkerboarding. Om en för hög penaliseringsfaktor ansätts konvergerar inte lösningen och därför rekommenderas att inte använda ett p större än 3 [4].



Figur 2.3: Elasticitetsmodul som funktion av densitet x_e och penaliseringsfaktorn p .

Det finns flera tillvägagångssätt för att lösa optimeringsproblemet i ekvation (2.5). Exempel på metoder är *Method of Moving Asymptots*, *Gradientmetoden* (eng. *Steepest Descent*) samt *Newtons metod* [3][4][6]. Den strategi som används i detta projekt, är den så kallade *Optimality Criteria Method*. Denna metod är en iterativ process som grundar sig på att lösa extremvärdesproblem för funktioner med hjälp av lagrangemultiplikatorer (eng. *Lagrange Multipliers*) [2][4]. För en introduktion och härledning till sistnämnda hänvisas läsaren till Adams [1]. *Optimality Criteria Method* grundar sig på att i varje iteration uppdatera elementdensiteterna enligt schemat som åskådliggörs i ekvation (2.8) [2].

$$x_e^{\text{new}} = \begin{cases} \max(0, x_e - m) & \text{om } x_e B_e^\eta \leq \max(0, x_e - m) \\ \min(1, x_e + m) & \text{om } x_e B_e^\eta \geq \min(1, x_e - m) \\ x_e B_e^\eta & \text{annars} \end{cases} \quad (2.8)$$

Här är x_e^{new} den uppdaterade densiteten, x_e den nuvarande densiteten, m en positiv stegkonstant och n en numerisk dämpningskoefficient. B_e erhålls från optimeringskriteriet enligt:

¹SIMP, Solid Isotropic Material with Penalization.

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}} \quad (2.9)$$

där lagrangemultiplikatorn λ beräknas med hjälp av bisektionsalgoritmen så att volymfraktionsvillkoret uppfylls [2]. Derivatorna av målfunktionen c och materialets volym V med avseende på elementdensiteterna x_e ges av ekvation (2.10). Det andra kriteriet i ekvation (2.10) utgår från antagandet att varje element har enhetsvolym [2].

$$\begin{cases} \frac{\partial c}{\partial x_e} = -px_e^{p-1} (E_0 - E_{\min}) \mathbf{u}_e^T \mathbf{K}_e^0 \mathbf{u}_e \\ \frac{\partial V}{\partial x_e} = 1 \end{cases} \quad (2.10)$$

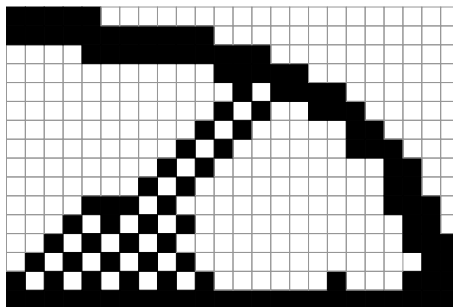
Efter varje uppdatering av designvariablerna enligt ekvation (2.8) kontrolleras kravet för konvergens. Kravet kan exempelvis vara att ändring av komplians eller densitet är tillräckligt liten.

2.3 Vanliga problem vid topologioptimering

Det finns ett antal vanliga komplikationer som ofta återkommer inom topologioptimering. Tre av de vanligaste problemen är checkerboarding, densitetsvariation och nätberoende. I detta kapitel beskrivs de tre problemen tillsammans med lösningsförslag.

2.3.1 Checkerboarding

Ett problem som ofta uppkommer vid topologioptimering är så kallad checkerboarding. Det innebär att lösningen som genereras innehåller regelbundna tomrum, som ses i Figur 2.4, och kan liknas vid ett schackbräde, därav termen checkerboarding. En sådan lösning är inte önskvärd eftersom den inte är realiserbar. Problemet är kopplat till den approximation som görs med hjälp av finita elementmetoden. När domänen diskretiseras med finita element överskattas styvheten diagonalt mellan element och checkerboardmönstret uppfattas som styvare än vad fallet egentligen är [3].



Figur 2.4: Illustration av checkerboarding, strukturen innehåller områden av material utplacerat i schackbrädesliknande mönster.

Det finns ett antal potentiella lösningar till detta problem. En lösning som visat sig fungera är att använda andra elementtyper. Checkerboarding förhindras i hög grad om rektangulära 8-nods eller 9-nodselement används istället för de vanligt använda 4-nodselementen [3]. Nackdelen med att använda element med fler antal noder är att beräkningstiden ökar avsevärt vilket kan bli problematiskt i takt med att antalet element ökar. Det är därför intressant att studera andra lösningar till checkerboarding som inte kräver lika mycket beräkningskapacitet.

Ytterligare en lösning för att komma tillrätta med checkerboarding är införandet av rotationsfrihetsgrader för elementen. De nya frihetsgraderna motverkar överskattandet av checkerboardingstrukturens styvhet genom att

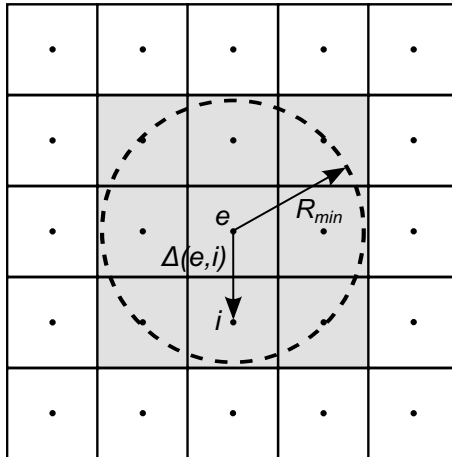
ta hänsyn till elementens rotation. Denna metod kan inte helt eliminera problemet med checkerboarding, men det reduceras avsevärt [7].

En vanlig metod för att lösa problemet med checkerboarding är att använda sig av olika typer av filter [3]. Ett filter modifierar lösningen i varje iteration och kan på detta sätt bidra till att lösa olika problem. Det så kallade känslighetsfiltret (*eng. sensitivity filter*) har visat sig vara effektivt mot checkerboarding. Filtret modifierar kompliansens derivator med avseende på designparametrarna [2][3]. Detta uppnås genom att för ett element, e , ta ett viktat medelvärde av de kringliggande elementens derivator, enligt [2]:

$$\left(\frac{\partial c}{\partial x_e}\right)^{\text{new}} = \frac{1}{\max(\alpha, x_e) \sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i \frac{\partial c}{\partial x_i} \quad (2.11)$$

där $\left(\frac{\partial c}{\partial x_e}\right)^{\text{new}}$ är den modifierade derivatan av kompliansen c med avseende på det aktuella elementets densitet, N_e är mängden av element i för vilka avståndet mellan respektive center, $\Delta(e, i)$, till element e är mindre än en filtreringsradie R_{\min} , vilket illustreras i Figur 2.5. En konstant med ett litet värde, α , ansätts för att undvika division med 0 och H_{ei} är en viktfaktor som definieras enligt:

$$H_{ei} = \max(0, R_{\min} - \Delta(e, i)) \quad (2.12)$$



Figur 2.5: Derivatan för elementet, e , i mitten modifieras med avseende på intilliggande (gråmarkerade) elements derivator.

Lösningar som genereras med hjälp av känslighetsfiltret innehåller ingen checkerboarding och påverkas ej av nätupplösningen. Det är dessutom enkelt att implementera och är tidseffektivt. Nackdelar med filtret är att det bildas stora densitetsvariationer i lösningen, och att lösningsrymden begränsas. Dessutom blir lösningen beroende av ytterligare en faktor, R_{\min} . Känslighetsfiltret är inte teoretiskt underbyggt och det är inte helt klarlagt varför det fungerar [3].

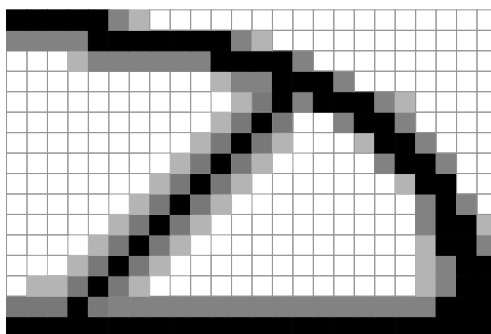
Ett annat filter är densitetsfiltret (*eng. density filter*). Filtret är snarlikt känslighetsfiltret, men här modifieras elementens designparametrar, x_e , direkt (istället för dess derivator). Den nya densiteten x_e^{new} fås som ett viktat medelvärde av de kringliggande elementens densiteter enligt [2]:

$$x_e^{\text{new}} = \frac{1}{\sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i \quad (2.13)$$

Resultaten som genereras med densitetsfiltret är närmast identiska med de som fås med känslighetsfiltret, och de har gemensamma för- och nackdelar [2].

2.3.2 Densitetsvariation

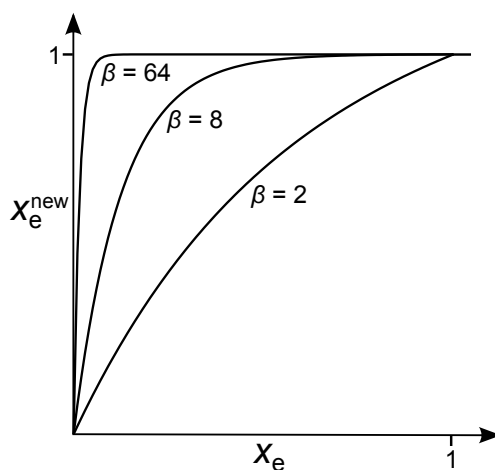
Densitetsvariationer i lösningen är ett problem som ofta uppkommer i enskilda delar av strukturen och vid användning av känslighets- respektive densitetsfilter, detta illustreras i Figur 2.6. Strukturer med varierande densiteter vill undvikas eftersom det av produktionsmässiga skäl är enklare att använda homogena material.



Figur 2.6: Illustration av densitetsvariationer där strukturens material inte är homogent.

Ett filter som inte ger upphov till lika mycket densitetsvariationer som ovanstående filter är det så kallade Heavisidefiltret. Detta bygger på samma metod som densitetsfiltret, dock modifieras designparametrarna ytterligare enligt ekvation (2.14) [2].

$$x_e^{\text{new}} = 1 - e^{-\beta x_e} + x_e e^{-\beta} \quad (2.14)$$

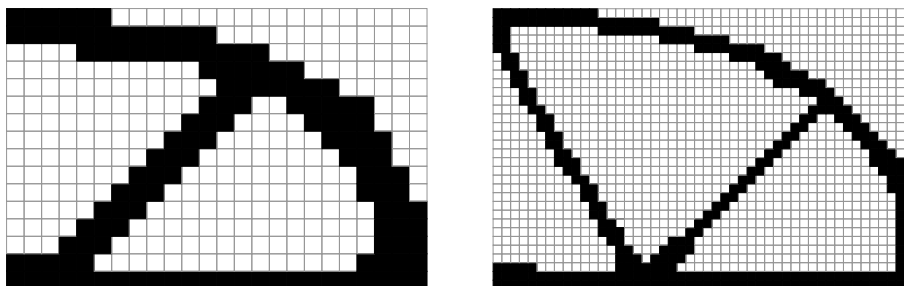


Figur 2.7: Ekvation (2.14) för olika värden på β .

Denna funktion går mot Heavisidefunktionen när faktorn β går mot oändligheten vilket gör att densiteterna modifieras till värden nära 1 eller 0. Faktorn ökas i takt med antalet iterationer och ger slutligen upphov till en svartvit lösning, det vill säga en struktur med enbart extrema värden på elementdensiteterna. En nackdel med filtret är att det påverkar volymfraktionen eftersom ekvation (2.14) gör att alla densiteter ökar, vilket kan ses i Figur 2.7. Detta leder till att β måste ökas successivt så att volymfraktionen kan justeras (enligt ekvation (2.8)) vilket förlänger den totala beräkningstiden [2].

2.3.3 Nätberoende

Nätberoende lösningar innebär att olika topologier genereras vid olika val av diskretisering (upplösning på elementnätet) vilket illustreras i Figur 2.8. När ett finare elementnät används bildas i allmänhet fler hål i strukturen. Om varje diskretisering ger olika topologier blir det svårt att bedöma vilken som är optimal att applicera i verkligheten.



Figur 2.8: Illustration av nätberoende, samma optimeringsproblem ger principiellt olika strukturer för två olika upplösningar.

Det finns olika förklaringar till uppkomsten av nätberoende resultat [3][4]. Den matematiska tolkningen är att topologi-optimeringsproblemet ekvation (2.5) i allmänhet saknar entydig lösning. Införandet av fler hål kommer enligt optimeringsproblemet (ekvation (2.5)) att öka effektiviteten av materialanvändandet. Den fysikaliska tolkningen är att sammansättningen av det tillgängliga materialet går mot en sorts optimal mikrostruktur [10].

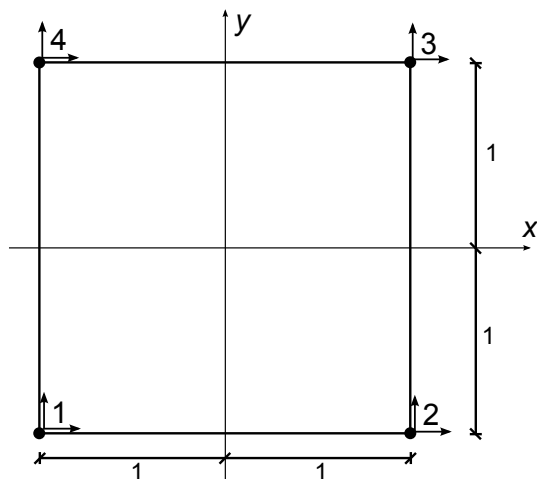
För att lyckas generera makroskopiska och nätberoende resultat måste lösningsrymden för möjliga strukturer begränsas. De metoder som används i dagsläget kan delas in i två klasser. Det första alternativet är att addera begränsningar till optimeringsproblemet, ekvation (2.5), vilket direkt reducerar antal möjliga lösningar. Det andra alternativet består av att implementera filtreringsalgoritmer under itereringsprocessen. Grundtanken för de flesta filtreringsalgoritmer som hanterar nätberoende lösningar är att begränsa strukturens totala omkrets, där den totala omkretsen definieras som summan av längderna för alla inre och yttre gränser. Genom att begränsa omkretsen reduceras markant antal hål som kan bildas [10].

2.4 Olika elementtyper

Det finns många sätt att diskretisera domänen över vilken FE-analysen görs. För- och nackdelar finns med varje elementtyp vilket gör att de lämpar sig för olika användningsområden. Element med fler frihetsgrader och noder ger möjlighet till en bättre approximation av lösningen till elasticitetsproblemet i ekvation (2.1). Fler frihetsgrader resulterar dock i ett större ekvationssystem vilket i sin tur ökar beräkningstiden. Detta leder ofta till att en avvägning behöver göras mellan noggrannhet och beräkningstid. Nedan följer en kortare beskrivning av de elementtyper som används i detta projektet.

2.4.1 Rektangulära element med fyra noder

Den elementtyp som används i top88 är ett rektangulärt element med fyra noder och bilineära basfunktioner (fortsättningsvis kallat Q4-element²), se Figur 2.9. Detta element har som fördel att vara mycket beräkningseffektivt på grund av att antalet frihetsgrader är lågt. I topologioptimering förknippas dock Q4-elementet med checkerboardproblemet. Basfunktionerna för elementet ges av ekvation (2.15) [9].



$$\begin{aligned}
 N_1 &= \frac{1}{4} (1 - x)(1 - y) \\
 N_2 &= \frac{1}{4} (1 + x)(1 - y) \\
 N_3 &= \frac{1}{4} (1 + x)(1 + y) \\
 N_4 &= \frac{1}{4} (1 - x)(1 + y)
 \end{aligned}
 \tag{2.15}$$

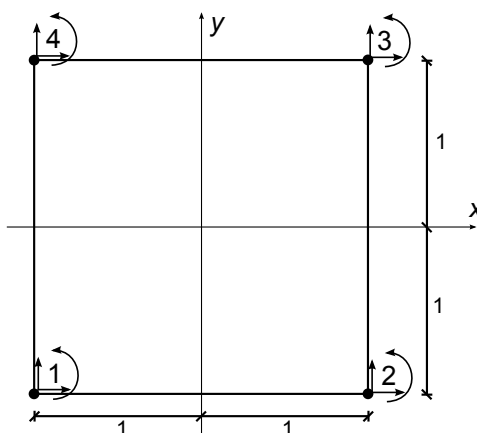
Figur 2.9: Q4-element med nodnumrering och frihetsgrader.

2.4.2 Rektangulära element med fyra noder och rotationsfrihetsgrader

En variant på Q4-elementet fås om en rotationsfrihetsgrad läggs till i varje nod. Utökningen av frihetsgrader ger en mer verklighetstrogen modellering, dock på bekostnad av beräkningstid. Element med rotationsfrihetsgrader, vilka i fortsättningen kallas Q4x-element, används ofta för att modellera skalstrukturer [7]. Med införandet av rotationsfrihetsgraderna krävs en ny formulering av elementstyvhetmatrisen. Enligt Ibrahimbegovic [5] kan styvhetmatrisen ställas upp enligt ekvation (2.16).

$$\mathbf{K} = \mathbf{K}_m + \gamma \mathbf{K}_{rot}
 \tag{2.16}$$

där \mathbf{K}_m beskriver translationsbidraget, \mathbf{K}_{rot} beskriver rotationsbidraget och γ är en problemberoende skalningsfaktor som relaterar rotationer till translationer. Det finns flera förslag på värdet för γ , men alla ligger i intervallet $G/1000$ och G där G är skjuvmodulen [7][8].

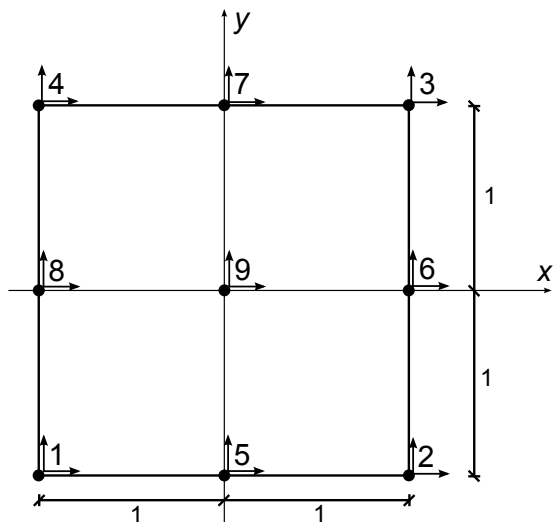


Figur 2.10: Q4x-element med 3 frihetsgrader i varje nod.

²Quadrilateral element with four nodes

2.4.3 Rektangulära element med nio noder

För att få en bättre FE-approximation kan en högre polynomapproximation användas. Ett vanligt sådant är ett rektangulärt element med 9 noder och kvadratiska basfunktioner enligt Figur 2.11 respektive ekvation (2.17). Denna elementtyp benämns i fortsättningen som Q9-element. Nackdelen med fler noder är återigen att beräkningstiden ökar [9].

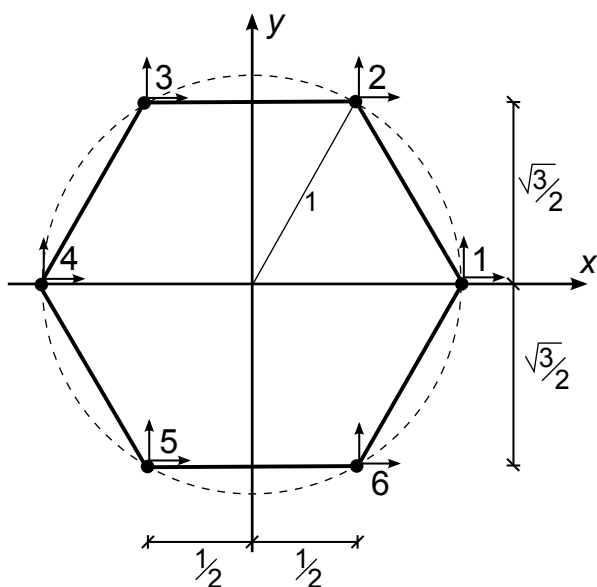


Figur 2.11: Q9-element med nodnumrering och frihetsgrader.

$$\begin{aligned}
 N_1 &= \frac{xy}{4}(x-1)(y-1) \\
 N_2 &= \frac{xy}{4}(x+1)(y-1) \\
 N_3 &= \frac{xy}{4}(x+1)(y+1) \\
 N_4 &= \frac{xy}{4}(x-1)(y+1) \\
 N_5 &= \frac{-y}{2}(y-1)(x^2-1) \\
 N_6 &= \frac{-x}{2}(x+1)(y^2-1) \\
 N_7 &= \frac{-y}{2}(x^2-1)(y+1) \\
 N_8 &= \frac{-x}{2}(x-1)(y^2-1) \\
 N_9 &= (x^2-1)(y^2-1)
 \end{aligned} \tag{2.17}$$

2.4.4 Hexagonala element med sex noder

I projektet har även hexagonelement använts (i fortsättningen kallad H6-element). Med hjälp av H6-element kan mer avancerade domäner byggas upp på grund av det större antalet sidor. En nackdel är att elementet har fler frihetsgrader och ger därför ett större ekvationssystem. Elementets geometri illustreras i Figur 2.12 och tillhörande basfunktioner, härledda av Tabarraei & Sukumar [11], ges av ekvation (2.18).



Figur 2.12: H6-element med nodnumrering och frihetsgrader.

$$\begin{aligned}
 N_1 &= \frac{8}{9}(1+x)\left(\frac{1}{2}+x\right)\left(\frac{1}{2}-x\right)\left(\frac{\sqrt{3}}{2}+y\right)\left(y-\frac{\sqrt{3}}{2}\right) \\
 N_2 &= \frac{8}{9}(1+x)(1-x)\left(\frac{1}{2}+x\right)y\left(\frac{\sqrt{3}}{2}+y\right) \\
 N_3 &= \frac{8}{9}(1+x)(1-x)\left(\frac{1}{2}-x\right)y\left(\frac{\sqrt{3}}{2}+y\right) \\
 N_4 &= \frac{8}{9}(1-x)\left(\frac{1}{2}+x\right)\left(\frac{1}{2}-x\right)\left(\frac{\sqrt{3}}{2}+y\right)\left(-\frac{\sqrt{3}}{2}+y\right) \\
 N_5 &= \frac{8}{9}(1+x)(1-x)\left(\frac{1}{2}-x\right)y\left(y-\frac{\sqrt{3}}{2}\right) \\
 N_6 &= \frac{8}{9}(1+x)(1-x)\left(\frac{1}{2}+x\right)y\left(y-\frac{\sqrt{3}}{2}\right)
 \end{aligned} \tag{2.18}$$

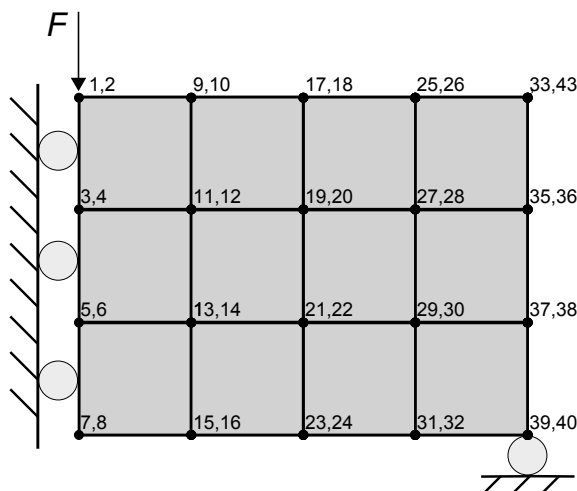
2.5 Topologioptimeringsprogrammet top88

Top88 är ett topologioptimeringsprogram skrivet i MATLAB och är ämnat för utbildningssyfte [2]. Programmet löser optimeringsproblemet för en MBB-balk³ där domänen diskretiseras med Q4-element. Av symmetriskäl modelleras endast halva balken, se Figur 2.13. Som indata kräver programmet sex olika parametrar och anropet ser ut enligt följande:

```
>> top88 (nelx, nely, volfrac, penal, rmin, ft)
```

där **nelx** och **nely** är antalet element i x- respektive y-riktningen, **volfrac** är volymfraktionen, **penal** är penaliseringsfaktorn för SIMP-metoden, **rmin** är den önskade filtreringsradien och **ft** väljs till 1 eller 2 beroende på om känslighetsfilter eller densitetsfilter skall användas.

Inledningsvis initieras FE-problemet genom att materialparametrar, elementstyvhetsmatris, lastvektor samt randvillkor definieras. Ett exempel kan ses i Figur 2.13 där topologin har definierats som en domän med 4x3 element. Elementens frihetsgrader numreras för x- och y-led för respektive nod enligt figuren. I det aktuella exemplet läses frihetsgrad 1, 3, 5, 7 samt 40 och kraften hamnar på frihetsgrad nummer 2.



Figur 2.13: Designomän med numrering av frihetsgrader för en halv MBB-balk.

När problemet är uppställt startar optimeringsloopen. Som startgissning används volymfraktionen, **volfrac**, som densitet för varje element. Varje iteration består av tre huvuddelar: FE-analys, filtrering samt optimering och uppdatering av densiteter. I FE-analysen ställs jämviktsekvationen $\mathbf{KU} = \mathbf{F}$ upp och förskjutningsvektorn \mathbf{U} räknas ut. Denna används sedan för att beräkna kompliansen och derivatorna av kompliansen med avseende på densiteterna. I nästa steg modifieras lösningarna med filter. Det finns två tillgängliga filter i top88, ett känslighetsfilter och ett densitetsfilter. Känslighetsfiltret modifierar derivatorna och densitetsfiltret modifierar densiteterna direkt enligt ekvation (2.11) respektive ekvation (2.13).

Den sista och viktigaste delen av varje iteration är uppdateringen av densiteterna. Detta sker med hjälp av lagrangemultiplikatorer och Optimality Criteria Method enligt ekvation (2.8). När de nya densiteterna är beräknade kontrolleras konvergenskravet, vilket i detta fall innebär att den största densitetsändringen för elementen är tillräckligt liten. Om lösningen inte har konvergerat krävs ytterligare iterationer.

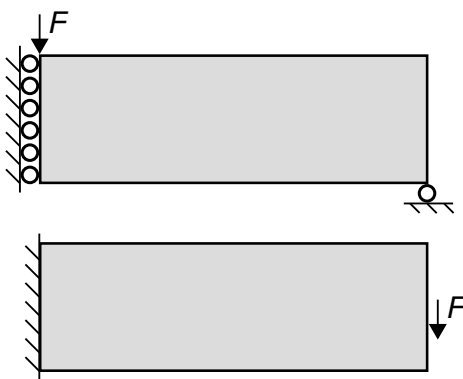
³Messerschmitt-Bölkow-Blohm-balk, fritt upplagd balk

3 METOD

För att utvärdera hur de olika lösningsmetoderna löser checkerboarding, densitetsvariationer och nätberoende ställdes fyra referensfall upp. Med hjälp av dessa genererades strukturer som sedan jämfördes med varandra. Referensfallen bidrar till att göra det enklare att jämföra och analysera de olika strukturerna. Vidare utformades kvantifierbara analysmetoder för numeriska jämförelser eftersom det inte fanns några, för författarna kända, etablerade standardmetoder för detta ändamål. Utöver de numeriska analysmetoderna studerades även strukturerna visuellt.

3.1 Referensfall

Referensfallen grundades på två olika lastfall. Det första lastfallet är en fritt upplagd balk med en vertikal punktlast i mitten. Av symmetriskäl modellerades bara halva balken enligt Figur 3.1. Det andra är en konsolbalk som belastas med en vertikal punktlast i den fria änden enligt Figur 3.1. Referensfallen utgår från dessa lastfall eftersom de är vanligt använda inom topologioptimering, se till exempel [2][3][4]. För att ytterligare öka kvaliteten av jämförelserna användes två olika nätupplösningar, 60x20 respektive 120x40 vilket totalt ger fyra olika referensfall.



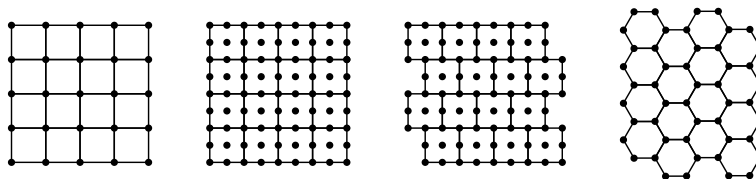
Figur 3.1: De två lastfallen som utgjorde referensfall. Överst en halv MBB-balk och underst en konsolbalk.

3.2 Implementerade lösningsförslag

De fyra referensfallen beskrivna ovan låg till grund för testerna av de olika lösningsförslagen som behandlades i detta projekt. Samtliga resultat genererades med SIMP-metoden med penaliseringsfaktorn $p = 3$ och en volymfraktion på 0,5. Materialparametrarna E_0 , E_{\min} och Poissons tal, ν , för framtagning av elementstyvhetsmatriser samt för ekvation (2.6) valdes till 1, 10^{-9} respektive 0,3. Samtliga elementtyper som beskrivs i kapitel 2.4, har testats för de fyra referensfallen. Motiveringen till varför dessa element studerades är:

- **Q4-element.** Det få antalet noder resulterar i korta simuleringstider vilket är intressant i industriella applikationer.
- **Q4x-element.** Om ett rotationsbidrag tillkommer i FE-modelleringen så förväntas checkerboardstrukturer inte vara lika motståndskraftiga mot rotation.
- **Q9-element.** Enligt litteraturen ska detta minska checkerboarding [3].
- **Förskjutna Q9-element.** Dessa element kommer alltid att dela minst två noder med samtliga angränsande element, och förväntas därför förhindra uppkomsten av checkerboarding genom att ett element då inte längre kan sitta ihop med endast en nod.
- **H6-element.** Samma motivering som för förskjutna Q9-element.

Konfigurationerna för de olika elementen kan ses i Figur 3.2. Notera att Q9-elementen har två olika konfigurationer. Q4x-elementen är beroende av en faktor gamma (se ekvation (2.16)), och tester har gjorts med olika värde på denna faktor.



Figur 3.2: Elementkonfigurationer som har studerats, Q4- och Q4x-, Q9-, förskjutna Q9- respektive H6-element.

De tre filtren, känslighets-, densitets- och Heavisidefiltret har undersökts. Valen av dessa motiveras av att de är vanligt använda inom topologioptimering och ger goda resultat [2]. Alla filter är beroende av en faktor R_{\min} och för att kunna studera inverkan av denna gjordes tester i vilka denna parameter varierades.

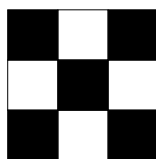
3.3 Analyismetoder

Vid analys av resultaten utvärderades fyra faktorer; beräkningstid, checkerboarding, densitetsvariation samt nätberoende. För att fastställa beräkningstider användes MATLABs tidtagningsfunktion `tic toc`. Detta moment har utförts på samma dator för att få en så rättvis mätning som möjligt. Tiden togs endast för lösandet av optimeringsproblemet eftersom andra faktorer såsom FE-initiering inte var av intresse att jämföra. För att ge en mer korrekt tidsbedömning togs medelvärde av 10 försök.

För att kunna göra en numerisk jämförelse av mängden checkerboarding i strukturerna införades ett mått, S , enligt:

$$S = \frac{n_S}{N} \quad (3.1)$$

där S är andelen checkerboardelement, n_S är antalet checkerboardelement och N är totala antalet element i domänen. Ett checkerboardelement definierades som ett ifyllt element med tomma angränsande element i x- och y-riktning se Figur 3.3. Ett tomt element definierades som ett element med densitet mindre än 0,1, och ett ifyllt element definierades som ett med densitet över 0,9. Element med densiteter mellan dessa gränsvärden klassificerades som intermediära element. Notera att denna definition av checkerboardelement ej är föreslagen i litteratur utan togs fram av författarna för att kvantifiera checkerboardproblemet.

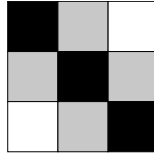


Figur 3.3: Illustration av ett checkerboardelement, ett element som är ifyllt och har tomma element i x- respektive y-led.

Vid analys av densitetsvariationer räknades antalet element med intermediära densiteter, alltså mellan 0, 1 och 0, 9, se Figur 3.4. Ett mått D infördes för att kvantifiera densitetsvariationen i en struktur enligt:

$$D = \frac{n_D}{N} \quad (3.2)$$

där D är andelen element med intermediära densiteter, n_D är antalet element med intermediära densiteter och N är totala antalet element i strukturen.



Figur 3.4: Illustration av densitetsvariation; 4 element med intermediär densitet (gråa).

Då det har varit svårt att kvantifiera nätberoende har denna faktor enbart analyserats visuellt. Anledningen till detta är att ingen etablerad metod för att kvantifiera nätberoende har påträffats, dessutom ansågs att en visuell jämförelse var tillräcklig.

3.4 Utformning av grafiskt gränssnitt

För att göra top88 användarvänligt utvecklades ett grafiskt gränssnitt för att på ett enkelt sätt kunna skapa och analysera olika domäner och lastfall i 2D. Det grafiska gränssnittet för programmet utformades med hjälp av MATLABs *GUIDE (Graphical User Interface Development Environment)*. Valet av *GUIDE* motiverades av att det är ett lätthanterligt verktyg för skapandet av professionella gränssnitt. Eftersom MATLAB användes för både top88 och skapandet av programmet blev interaktionen mellan dessa naturlig. Programmet kommer i fortsättningen att kallas för *topGUI*.

För att topGUI skulle bli användarvänligt och flexibelt med lastfall och parametrar utformades gränssnittet med följande funktioner:

- **Definera domän.** En godtycklig domän ska kunna defineras över vilken optimeringen sker på.
- **Definera stöd.** Stöd som låser frihetsgrader ska kunna placeras godtyckligt.
- **Definera krafter.** Krafter ska kunna placeras med godtycklig riktning och magnitud.
- **Analys.** TopGUI ska kunna visualisera den framtagna strukturens utböjning och spänningar.
- **Spara lastfall.** Användaren ska kunna ha möjlighet att kunna spara och återanvända lastfall.

TopGUI grundar sig på samma optimeringsalgoritm som i top88, dock har koden för FE-initiering omstrukturerats då den befintliga koden inte fungerar för godtyckliga lastfall och domäner.

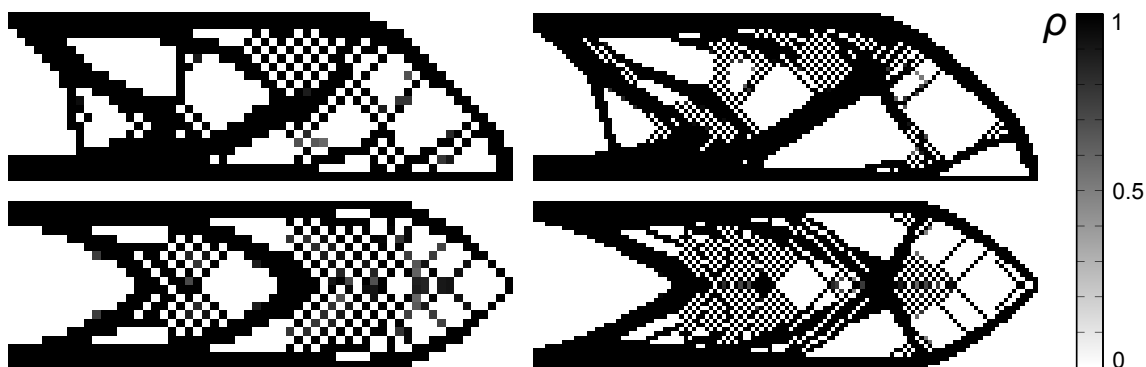
4 RESULTAT

Under projektets gång har flera olika lösningsförslag utvärderats. Resultaten som presenteras är indelade i tre kategorier. Inledningsvis presenteras data från lösningsförslag då elementtyper och filter har implementerats. Därefter sammanställs resultaten i en jämförande tabell. De analysmetoder som använts för att kvantifiera problemen; checkerboarding och densitetsvariationer, är beskrivna i kapitel 3.3. Slutligen beskrivs möjligheterna med det utvecklade topologioptimeringsprogrammet topGUI.

4.1 Olika elementtyper

4.1.1 Rektangulära element med fyra noder

De strukturella lösningar som fås när Q4-element används i FE-diskretiseringen återfinns, för samtliga referensfall, i Figur 4.1. Det mest framträdande av de undersökta problemen är checkerboarding, vilket uppgår till 5,7 procent i fallet med konsolbalk med nätupplösning 60x20. Även de andra fallen ger värden i samma storleksordning, vilka kan ses i Tabell 4.1. Problemet med nätberoende återfinns för båda fallen då endast de grövsta dragen återkommer för båda upplösningarna. Strukturer baserade på Q4-element uppvisar inga stora densitetsvariationer, på grund av SIMP. Det högsta värdet återfinns hos konsolen med upplösning 60x20, vilken har en densitetsvariation på 4,3 procent.



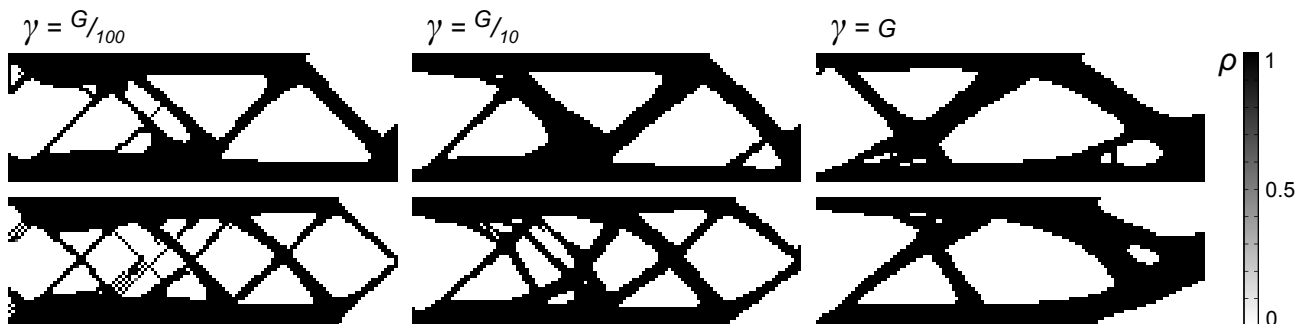
Figur 4.1: Optimerade strukturer med Q4-element för MBB-balk (överst) och konsolbalk (nederst) med nätupplösningar 60x20 (vänster) respektive 120x40 (höger).

Tabell 4.1: Andelen checkerboarding samt densitetsvariationer för strukturerna i Figur 4.1

	MBB 60x20	MBB 120x40	Konsol 60x20	Konsol 120x40
Checkerboarding	0,043	0,032	0,057	0,050
Densitetsvariation	0,010	0,006	0,043	0,008

4.1.2 Rektangulära element med fyra noder och rotationsfrihetsgrader

De resultat som har fåtts när Q4x-element används kan ses i Figur 4.2. De numeriska värdena för denna elementtyp är bra, vilka ses i Tabell 4.2. Lösningen innehåller endast måttlig checkerboarding och densitetsvariationer uppträder ej. Värdet på γ påverkar lösningen i stor grad och det är svårt att bedöma vilket värde som ger bäst resultat. Svårigheten ligger i att veta vilket γ som ger mest verklighetstrogen modellering.



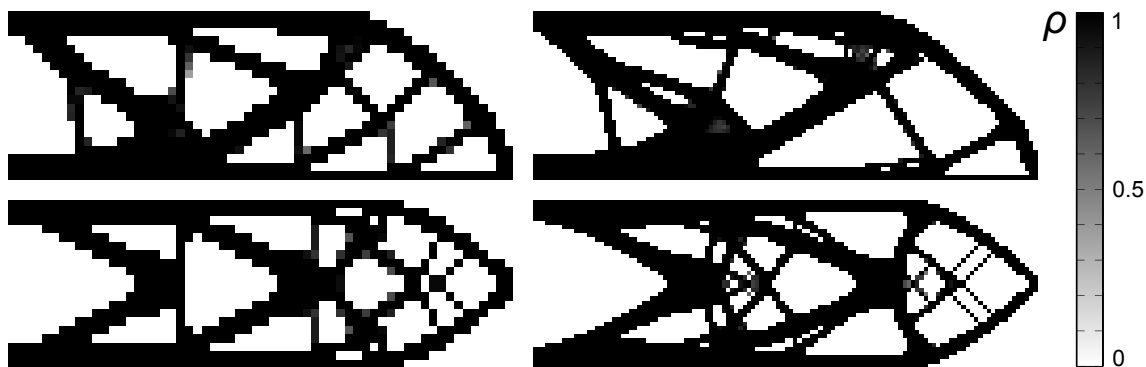
Figur 4.2: Optimerade strukturer för nätupplösning 120×40 med $Q4x$ -element för MBB-balk (överst) och konsolbalk (nederst) med $\gamma = G/100$ (vänster), $\gamma = G/10$ (mitten) och $\gamma = G$ (höger).

Tabell 4.2: Checkerboarding och densitetsvariation för strukturerna i Figur 4.2

Checkerboarding	$\gamma = G/100$	$\gamma = G/10$	$\gamma = G$
MBB 120×40	0,001	0,000	0,000
Konsol 120×40	0,009	0,000	0,000
Densitetsvariation			
MBB 120×40	0,000	0,000	0,000
Konsol 120×40	0,001	0,000	0,000

4.1.3 Rektangulära element med nio noder

Då $Q9$ -element används genereras lösningar som är av hög kvalitet med avseende på två av de karaktäristiska problemen som undersökts i projektet. Strukturerna, som kan ses i Figur 4.3, uppvisar inte någon checkerboarding och låga värden på densitetsvariationer, vilket kan ses i Tabell 4.3. Problemet med nätberoende syns dock tydligt. Utifrån resultaten som presenteras i 4.3 kan ses att de grövre dragen för de båda nätupplösningarna är lika, men då en finare diskretisering används fås tunnare delar och fler hål i strukturen. Resultatet är genomgående både för MBB-balken och för konsol-balken.



Figur 4.3: Optimerade strukturer med $Q9$ -element för MBB-balk (överst) och konsolbalk (nederst) med nätupplösningar 60×20 (vänster) respektive 120×40 (höger).

Tabell 4.3: Andelen checkerboarding samt densitetsvariationer för strukturerna i Figur 4.3

	MBB 60×20	MBB 120×40	Konsol 60×20	MBB 60×20
Checkerboarding	0,000	0,000	0,003	0,005
Densitetsvariation	0,019	0,010	0,015	0,008

4.1.4 Förskjutna Rektangulära element med nio noder

Lösningarna som genererats med den alternativa uppställningen av Q9-elementen uppvisar tydlig likhet med motsvarande resultat för originalkonfigurationen av Q9-elementen, vilket kan ses i Figur 4.4. Strukturerna innehåller fortfarande måttliga densitetsvariationer samt uppvisar nätberoende, dock är checkerboarding eliminerat vilket ses i Tabell 4.4.



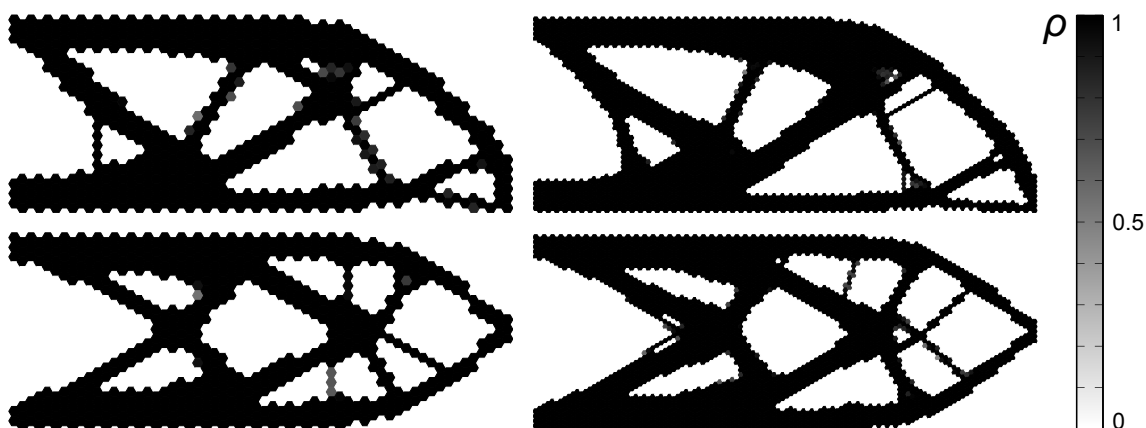
Figur 4.4: Optimerade strukturer med förskjutna Q9-element för MBB-balk (överst) och konsolbalk (nederst) med nätupplösningar 60x20 (vänster) respektive 120x40 (höger).

Tabell 4.4: Andelen checkerboarding samt densitetsvariationer för strukturerna i Figur 4.4

	MBB 60x20	MBB 120x40	Konsol 60x20	Konsol 120x40
Checkerboarding	0,000	0,000	0,000	0,000
Densitetsvariation	0,013	0,007	0,012	0,011

4.1.5 Hexagonala element med sex noder

Strukturer som genereras med H6-element uppvisar mycket goda resultat, vilket illustreras i Figur 4.5. Lösningarna innehåller ingen checkerboarding och endast försumbara densitetsvariationer, vilka redovisas i Tabell 4.5. H6-element är den elementtyp som uppvisar minst nätberoende då de principiella geometrierna är mycket lika för de olika nätupplösningarna.



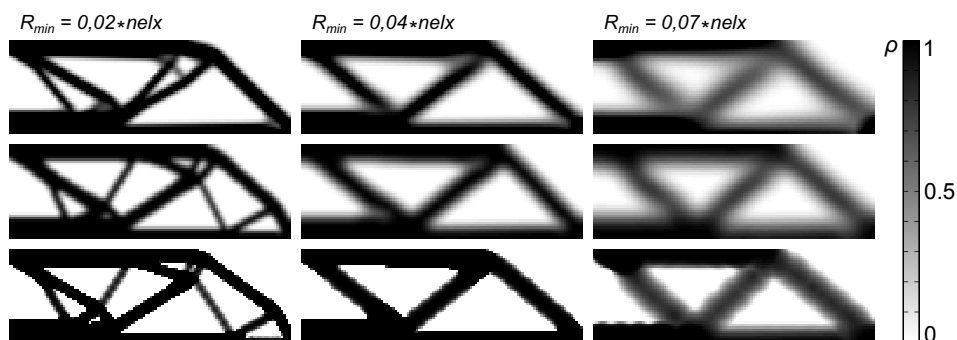
Figur 4.5: Optimerade strukturer med H6-element för MBB-balk (överst) och konsolbalk (nederst) med nätupplösningar 60x20 (vänster) respektive 120x40 (höger).

Tabell 4.5: Andelen checkerboarding samt densitetsvariationer för strukturerna i Figur 4.5

	MBB 60x20	MBB 120x40	Konsol 60x20	Konsol 120x40
Checkerboarding	0,000	0,000	0,000	0,000
Densitetsvariation	0,015	0,005	0,004	0,006

4.2 Filter

Strukturer som genererats med Q4-element och olika typer av filter kan ses i Figur 4.6 för referensfallet med MBB-balken. Samtliga resultat uppvisar mycket stora densitetsvariationer, dock har checkerboarding helt eliminerats, vilket ses i Tabell 4.6. När nätberonde studerades, se Bilaga A, drogs slutsatsen att lösningarna uppvisar identiska strukturer och filter kan därför antas vara oberoende av vald nätupplösning. Med ett lägre värde på R_{\min} modifieras derivatorna (av kompliansen med avseende på densiteten), alternativt densiteten inte tillräckligt. Med ett högre värde påverkas parametrarna för mycket och materialet blir utsmetat.

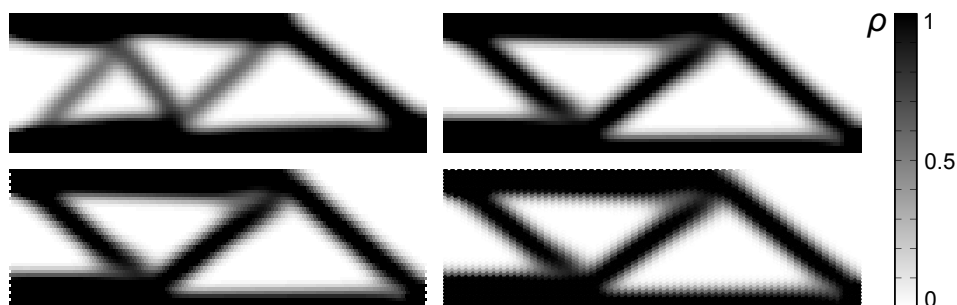


Figur 4.6: Optimerade strukturer genererade med Q4-element och känslighetsfilter (överst), densitetsfilter (mitten) och heavisidefilter (underst) för olika värden på R_{\min} .

Tabell 4.6: Andelen element med intermediära densiteter för strukturerna i Figur 4.6

	$R_{\min} = 0.02 * nelx$	$R_{\min} = 0.04 * nelx$	$R_{\min} = 0.07 * nelx$
Känslighetsfilter	0,21	0,32	0,65
Densitetsfilter	0,29	0,40	0,66
Heavisidefilter	0,07	0,07	0,44

Lösningarna i Figur 4.7 har genererats med filter i kombination med andra elementtyper. För Q4x-, Q9-, förskjutna Q9- och H6-element erhöles principiellt samma resultat. Resultatet är alltså oberoende av vald elementtyp eftersom filtrets egenskaper dominerar.



Figur 4.7: Optimala strukturer genererade med känslighetsfilter för Q4x- (uppe vänster), Q9- (uppe höger), förskjutna Q9- (nere vänster) och H6-element (nere höger).

4.3 Jämförelse mellan lösningsmetoder

Beräkningstiden för de olika lösningsförslagen presenteras i Tabell 4.7. I tabellen har tiderna normerats mot tiden för Q4-element som valdes som referens. Som väntat tar varje iteration längre tid ju fler frihetsgrader elementen har; H6-elementen med 12 frihetsgrader tar 2,6 gånger längre tid än Q4-elementen och Q9-elementet med 18 frihetsgrader tar 5,3 gånger så lång tid. Värt att notera är att filtren knappt påverkar beräkningstiden och är alltså en tidseffektiv lösningsmetod.

Tabell 4.7: Normaliserade tider för olika element och filter för de fyra referensfallen

	Q4	Q4x ($\gamma = G/10$)	Q9	förskjutna Q9	H6	Känslig- hets	Dens- itets	Heavi- side
MBB 60x20	1,0	2,1	5,2	5,9	2,5	1,0	1,1	1,2
Konsol 60x20	1,0	2,2	5,5	6,1	2,5	1,0	1,1	1,2
MBB 120x40	1,0	2,0	5,1	5,8	2,6	1,0	1,3	1,4
Konsol 120x40	1,0	2,1	5,3	5,9	2,7	1,0	1,3	1,4
Medel	1,0	2,1	5,3	5,9,	2,6	1,0	1,2	1,3

För att enklare få en överblick av de olika metoderna sammanställs resultaten i Tabell 4.8. Resultaten presenteras endast för MBB-balken med nätupplösningen 120x40 då de anses vara representativa för de andra referensfallen. Tabeller för de övriga referensfallen återfinns i Bilaga B.

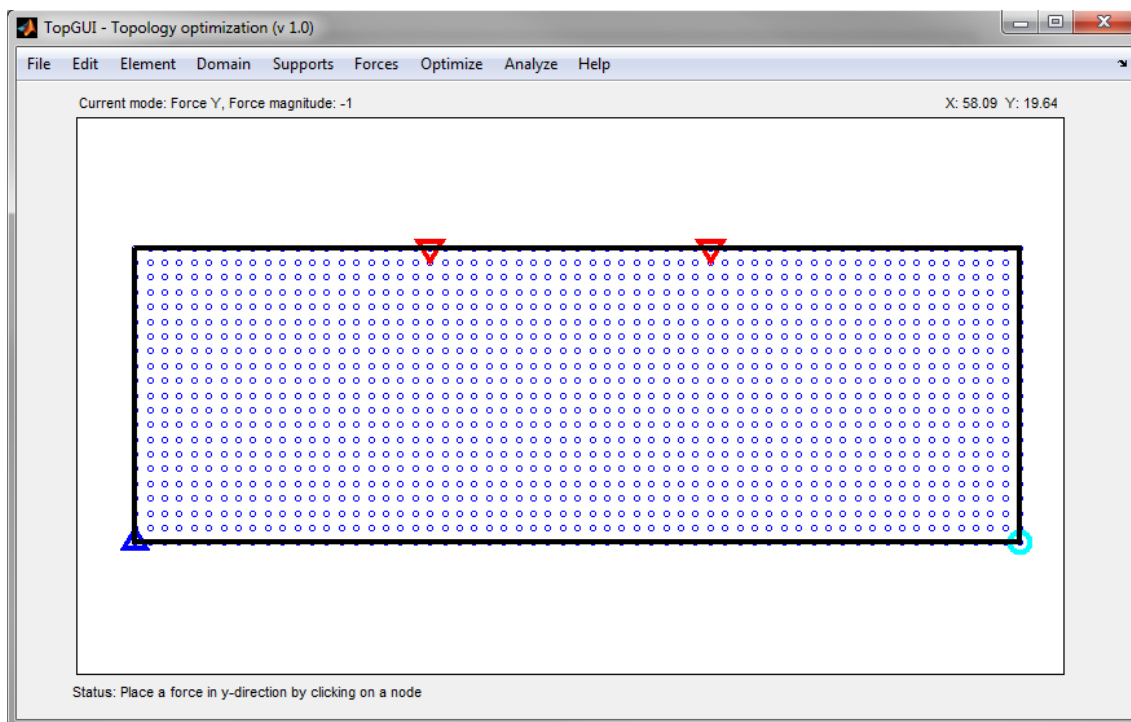
Tabell 4.8: Sammanställd data för referensfallet MBB-balk 120x40

	Relativ tid	Checkerboarding	Densitetsvariation
Q4	1,000	0,032	0,006
Q4x ($\gamma = G/10$)	1,978	0,000	0,000
Q9	5,147	0,000	0,010
Förskjutna Q9	5,753	0,000	0,010
H6	2,602	0,000	0,005
Känslighetsfilter	1,002	0,000	0,316
Densitetsfilter	1,302	0,000	0,395
Heavisidefilter	1,404	0,000	0,066

Checkerboarding återfinns endast för Q4- och Q4x element. Övriga lösningsmetoder hanterar detta problem bra. Densitetsvariationer uppkommer endast i betydande grad när filter används. För de andra metoderna är införandet av penaliseringsfaktorn p för SIMP tillräcklig för att ge en svartvit struktur.

4.4 Grafiskt gränssnitt för topologioptimering

Det slutgiltiga designen för topGUI kan ses i Figur 4.8. Programmet fungerar tillsammans med elementtyperna Q4, Q9 och H6. I fallet med H6 är dock användningsområdet begränsat då användaren inte har möjlighet att definera en godtycklig domän. För att starta optimeringen krävs att en domän med randvillkor och laster har ritats upp. Programmet optimerar sedan en struktur, baserat på styvhet, vilken kan analyseras med avseende på utböjning och spänning.



Figur 4.8: Domän med utplacerade stöd (blåa markeringar) och krafter (röda markeringar).

För att visa på programmets möjligheter har ett lastfall ställts upp och topGUI har sedan utfört en optimering för fallet. Då topGUI:s sekundära funktion är att analysera spänningar och utböjningar har detta utförts på strukturen och presenteras i Figur 4.9. Elementtypen som har använts i detta fall är Q9.



Figur 4.9: Demonstration av topGUI. Från vänster visas först domänen med stöd och krafter, sedan den optimerade strukturen, sedan totala utböjningar och till sist effektivspänningar enligt von Mises.

5 DISKUSSION OCH SLUTSATSER

I detta projekt har det tydligt visats att strukturer som genereras utifrån ett lastfall kan anta många olika former. Lösningmetoderna som har utvärderats i projektet har visat på olika resultat gällande hur effektivt de löser de karaktäristiska problemen; checkerboarding, densitetsvariation och nätberoende, som starkt förknippas med topologioptimering.

Checkerboarding är det problem som haft störst fokus i projektet. Problemet har stark koppling till Q4-elementen och är inte något som är utmärkande för de övriga lösningmetoderna. Från detta resultat dras slutsatsen att Q4-element inte är lämpliga att användas utan modifiering. Istället bör de användas tillsammans med ett filter som förhindrar uppkomsten av checkerboarding. Om användning av filtreringsalgoritmer vill undvikas har det i projektet visat sig effektivt att använda andra typer av element. Q4x-, Q9- och H6-element har alla gett goda resultat med avseende på checkerboardproblematiken.

Ytterligare ett komplikation som studerats är problemet med intermediära densiteter i den optimerade strukturen. I projektet har det framgått att dessa densitetsvariationer är starkt kopplade till filtreringsalgoritmerna. Eftersom både känslighetsfiltret och densitetsfiltret ger stora variationer kan heavisidefiltret anses bäst i detta avseende. Utan filter blir strukturerna i stort sett fria från densitetsvariationer vilket är en direkt följd av penaliseringsfaktorn p i SIMP-metoden.

Det tredje problemet som har varit i fokus i projektet är de genererade strukturernas beroende av vald nätupplösning. Resultaten visar att det främst är de tre olika filtren som gör att problemet kan undvikas, men även H6-elementen uppvisar minimal variation mellan olika nätupplösningar. För strukturer genererade med Q9-element kan specifika drag ses för både en grövre och finare nätupplösning, men strukturerna skapade med den finare nätupplösningen har tunnare delar och fler hål. Strukturer genererade med Q4-element visar på ett starkt beroende av nätupplösningen. Q4-elementen är därmed även utifrån problemet med nätberoende ett sämre alternativ vid topologioptimering, om det inte kombineras med filtreringsalgoritmer.

Utifrån undersökningarna som genomförts i projektet har det visat sig att de tre problemen; checkerboarding, densitetsvariation varandra. Exempelvis löser filtreringsalgoritmerna problemen med checkerboarding och nätberoende på ett tillfredsställande sätt men ger dessvärre stora densitetsvariationer.

I de fall filtreringsalgoritmer är implementerat i optimeringsprocessen har samtliga resultat uppvisat liknande strukturer. De filter som använts modifierar lösningen kraftigt genom att begränsa lösningrymden för den optimerade strukturen. En naturlig fråga som uppstår är i vilken omfattning filtret påverkar lösningen och om den faktiskt är optimal. Fördelen med filter är att de ger enklare strukturer som är enklare att producera. Eftersom det principiella slutresultatet ej påverkas av vilken elementtyp som används tillsammans med filtret anses Q4-elementet vara bäst lämpat, eftersom beräkningstiden minimeras.

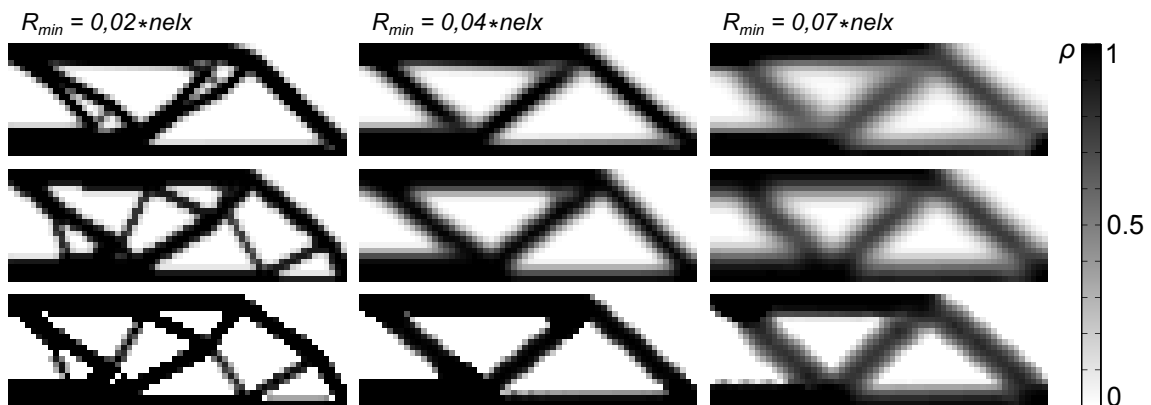
Då den aktuella topologioptimeringen sker med avseende på komplians behövs ofta ytterligare analys i termer av något som direkt kan relateras till materialets mekaniska prestanda. Skulle hänsyn ej tas till detta finns en risk att till exempel spänningskoncentrationer kan uppträda i utsatta delar av strukturen. En möjlig lösning på detta är att inkludera krav på maximal spänning direkt i optimeringsproblemet. Detta kan leda till att vidareutvecklingen av strukturen inte blir lika omfattande. Med dagens teknik krävs det i allmänhet efterkonstruktioner av olika slag även om hänsyn har tagits till andra faktorer än komplians. Till exempel krävs det ofta att strukturen rationaliseras för att enkelt kunna produceras. Nackdelen med efterkonstruktioner är att det kräver både tid och kompetens. Dessutom frångås den styvaste lösningen när den modifieras i efterhand.

Topologioptimering ger stora möjligheter till att effektivisera produktutvecklingsfasen. Förhoppningen är att i framtiden kunna få färdiga strukturer direkt, utan att behöva göra förändringar i efterhand. Inte minst ses stora framtida möjligheter tillsammans med utvecklingen av 3D-skrivare där det är av intresse att minska materialåtgången. Då kunskapen i ämnet stadigt växer och den tekniska utvecklingen går framåt kommer intresset, användningen och möjligheterna med topologioptimering att öka.

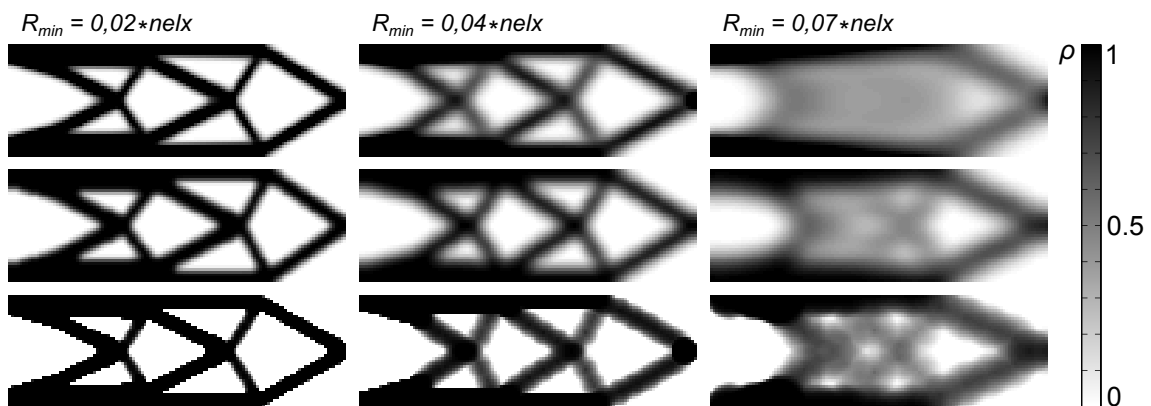
Referenser

- [1] R. Adams och C. Essex. *Calculus - A Complete Course*. 7th edition. Pearson Canada Inc., 2010. ISBN: 978-0-321-54928-0.
- [2] E. Andreassen m. fl. Efficient topology optimization in MATLAB using 88 lines of code. *Springer* (2010).
- [3] M. Bendsoe och O. Sigmund. *Topology Optimization - Theory, Methods and Applications*. 2nd edition. Springer-Verlag, 2004. ISBN: 978-3-642-07698-5.
- [4] P. W. Christensen och A. Klarbring. *An Introduction to Structural Optimization*. Vol. 153. Springer Science + Business Media B.V., 2009. ISBN: 978-1-4020-8665-6.
- [5] A. Ibrahimbegovic, R. Taylor och E. Wilson. A Robust Quadrilateral Membrane Finite Element With Drilling Degrees of Freedom. *International Journal for Numerical Methods in Engineering* (1990).
- [6] H. Johansson. "Steepest descent and Newton methods using duality in Structural Optimization". Examensarb. Chalmers University of Technology, 2003.
- [7] C. Long. "Finite element developments and applications in structural topology optimization". Diss. University of Pretoria, 2007.
- [8] H. Nguyen-Van, N. Mai-Duy och T. Tran-Cong. An improved quadrilateral at element with drilling degrees of freedom for shell structural analysis. *Springer* (2010).
- [9] N. Ottosen och H. Petersson. *Introduction to the Finite Element Method*. Prentice Hall, 1992. ISBN: 978-0134738772.
- [10] O. Sigmund och J. Petersson. Numerical instabilities in topology optimization. *International Journal for Numerical Methods in Engineering* (1998).
- [11] A. Tabarraei och N. Sukumar. International Journal of Computational Methods. *Application of Polygonal Finite Elements in Linear Elasticity* (2006).

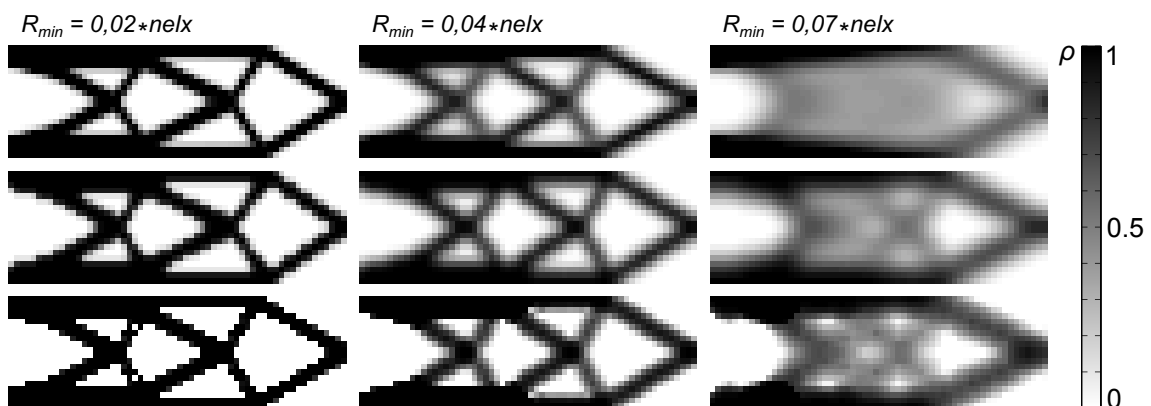
A Ytterligare strukturer med filter och varierande R_{min}



Figur A.1: Strukturer genererade med känslighetsfilter för en MBB-balk med upplösning 60×20



Figur A.2: Strukturer genererade med känslighetsfilter för en konsol-balk med upplösning 120×40



Figur A.3: Strukturer genererade med känslighetsfilter för en konsol-balk med upplösning 60×20

B Sammanställda tabeller för samtliga referensfall

Tabell B.1: Sammanställd data för referensfallet MBB-balk 60x20

	Relativ tid	Checkerboarding	Densitetsvariation
Q4	1,000	0,043	0,010
Q4x ($\gamma = G/10$)	2,076	0,000	0,000
Q9	5,327	0,000	0,019
Förskjutna Q9	5,887	0,000	0,011
H6	2,455	0,000	0,015
Känslighetsfilter	1,014	0,000	0,306
Densitetsfilter	1,075	0,000	0,383
Heavisidefilter	1,175	0,000	0,130

Tabell B.2: Sammanställd data för referensfallet konsol-balk 60x20

	Relativ tid	Checkerboarding	Densitetsvariation
Q4	1,000	0,057	0,043
Q4x ($\gamma = G/10$)	2,157	0,000	0,000
Q9	5,576	0,003	0,015
Förskjutna Q9	6,081	0,000	0,012
H6	2,512	0,000	0,004
Känslighetsfilter	1,034	0,000	0,513
Densitetsfilter	1,130	0,000	0,528
Heavisidefilter	1,180	0,000	0,335

Tabell B.3: Sammanställd data för referensfallet MBB-balk 120x40

	Relativ tid	Checkerboarding	Densitetsvariation
Q4	1,000	0,032	0,006
Q4x ($\gamma = G/10$)	1,978	0,000	0,000
Q9	5,147	0,000	0,010
Förskjutna Q9	5,753	0,000	0,010
H6	2,602	0,000	0,005
Känslighetsfilter	1,002	0,000	0,316
Densitetsfilter	1,302	0,000	0,395
Heavisidefilter	1,404	0,000	0,066

Tabell B.4: Sammanställd data för referensfallet konsol-balk 120x40

	Relativ tid	Checkerboarding	Densitetsvariation
Q4	1,000	0,050	0,008
Q4x ($\gamma = G/10$)	2,058	0,000	0,000
Q9	5,368	0,005	0,008
Förskjutna Q9	5,887	0,000	0,011
H6	2,651	0,000	0,006
Känslighetsfilter	1,015	0,000	0,509
Densitetsfilter	1,331	0,000	0,526
Heavisidefilter	1,448	0,000	0,277