



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STAVEBNÍ
ÚSTAV STAVEBNÍ MECHANIKY

FACULTY OF CIVIL ENGINEERING
INSTITUTE OF STRUCTURAL MECHANICS

STUDIE NÁVRHU VHODNÉHO TVARU MEMBRÁNOVÝCH KONSTRUKCÍ

STUDY OF DESIGN OF A SUITABLE SHAPE OF MEMBRANE STRUCTURES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. KAREL KOCINA

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. IVAN NĚMEC, CSc.

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA STAVEBNÍ

Studijní program	N3607 Stavební inženýrství
Typ studijního programu	Navazující magisterský studijní program s prezenční formou studia
Studijní obor	3607T009 Konstrukce a dopravní stavby
Pracoviště	Ústav stavební mechaniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Diplomant	Bc. Karel Kocina
Název	Studie návrhu vhodného tvaru membránových konstrukcí
Vedoucí diplomové práce	doc. Ing. Ivan Němec, CSc.
Datum zadání diplomové práce	31. 3. 2011
Datum odevzdání diplomové práce	13. 1. 2012
V Brně dne 31. 3. 2011	

.....
prof. Ing. Drahomír Novák, DrSc.
Vedoucí ústavu

.....
prof. Ing. Rostislav Drochytka, CSc.
Děkan Fakulty stavební VUT

Podklady a literatura

Němec et al. Finite Element Analysis of Structures

Zásady pro vypracování

Studie musí obsahovat:

1. zhodnocení současného stavu návrhu tvaru membrán pomocí specializovaných programů Formfinder a Rhino.
2. Prověření možností návrhu tvaru membrán pomocí programu RFEM.
3. Doporučení k vývoji specializovaného modulu programového systému RFEM pro návrh tvaru membrán.

Předepsané přílohy

Licenční smlouva o zveřejňování vysokoškolských kvalifikačních prací

.....
doc. Ing. Ivan Němec, CSc.
Vedoucí diplomové práce

Abstrakt

Tato diplomová práce se zabývá metodami návrhu tvaru membránových konstrukcí. Náplní této práce je analýza návrhů tvaru programy FormFinder a Rhinoceros v programu RFEM a porovnání výsledků. Dále pak otestovat možnosti návrhu tvaru programem RFEM.

Klíčová slova

hledání tvaru, metoda hustoty síly, dynamická relaxace, strategie aktualizované referenční plochy, membránové konstrukce

Abstract

This diploma thesis deals with methods for design of membrane structure shape. Main purpose is to analyze topology designs by Formfinder and Rhinoceros in RFEM and compare results. Test a possibility of designing shape by software RFEM.

Keywords

Form-Finding, Force Density Method, Dynamic Relaxation, Updated Reference strategy, Membrane Structures

Bibliografická citace VŠKP

KOCINA, Karel. *Studie návrhu vhodného tvaru membránových konstrukcí*. Brno, 2011. 87 s., 32 s. příl. Diplomová práce. Vysoké učení technické v Brně, Fakulta stavební, Ústav stavební mechaniky. Vedoucí práce doc. Ing. Ivan Němec, CSc..

Prohlašuji, že jsem diplomovou práci vypracoval samostatně pod vedením doc. Ing. Ivana Němce, Csc. a uvedl jsem v seznamu literatury všechny použité odborné zdroje.

V Brně dne 13. ledna 2012

Bc. Karel Kocina

Poděkování:

Děkuji svému vedoucímu diplomové práce doc. Ing. Ivanovi Němcovi, Csc. za více než vstřícný přístup k vedení a velikou pomoc při vypracování mé diplomové práce.

Dále bych rád poděkoval firmě Ing. Software Dlubal, s.r.o. za zapůjčení studentské licence k programu RFEM 4, bez kterého bych nemohl svoji diplomovou práci vypracovat.

Nejvíce ze všech bych však chtěl poděkovat svojí snoubence Ing. Pavle Juračkové, za poskytnutí perfektního zázemí a stálé psychické podpory. Jen díky ní se mi podařila má diplomová práce dokončit.

Rovněž děkuji všem, kteří, ať již přímo (dobrou radou nebo připomínkou) nebo nepřímo, přispěli ke zdárnému vypracování této práce.

V Brně dne 13. ledna 2012

OBSAH

1. ÚVOD.....	7
2. FORM-FINDING	8
2.1 Historický vývoj	8
2.2 Metody pro hledání tvaru	9
2.2.1 Metoda Force Density	9
2.2.2 Metoda Dynamické relaxace [1].....	9
2.2.3 Metoda Updated reference strategy [2].....	11
2.2.4 Nelineární analýza [3]	14
2.3 Programy na hledání vhodného tvaru	19
2.3.1 Form Finder	19
2.3.2 RhinoMembrane	19
2.3.3 Skript na relaxaci	20
2.3.4 RFEM.....	21
2.4 NURBS křivky a plochy	22
3. ANALÝZA VYBRANÝCH TVARŮ V PROGRAMU RFEM.....	24
3.1 Tvary vytvořené pomocí Rhino skriptů.....	24
3.1.1 Rotační plocha	24
3.1.2 Čtvercový půdorys	34
3.1.3 Shrnutí	43
3.2 Tvary vytvořené v programu Formfinder	44
3.2.2 Shrnutí	53
3.3 Hledání tvaru programem RFEM	54
3.3.1 Vynucený posun podpor	54
3.3.2 Předpětí v ploše pomocí smrštění.....	58
3.3.3 Shrnutí	84
4. ZÁVĚR.....	85
5. LITERATURA.....	86
6. SEZNAM PŘÍLOH.....	87

1. ÚVOD

Tato diplomová práce se zabývá hledáním vhodného tvaru membránových konstrukcí. Membránové konstrukce jsou vhodné zejména tam, kde je požadavek na transparentní, lehkou konstrukci, a proto jsou využívány především pro zastřešení venkovních prostorů jako jsou atria, terasy, stadiony apod.

Teoretická část diplomové práce shrnuje informace o procesu hledání tvaru, jeho historii, metodách a programech sloužících k jeho návrhu. Konkrétně se jedná o programy Rhinoceros, FormFinder a RFEM.

V praktické části diplomové práce jsou navrženy různé tvary membrán s použitím programů uvedených v teoretické části. Tyto tvary jsou následně přeneseny do programu RFEM od firmy Ing. Software Dlubal, s.r.o. a analyzovány. Podstatnou část tvoří návrh vhodného tvaru přímo pomocí programu RFEM s využitím jeho potenciálu při řešení nelineárních úloh. V závěru této kapitoly je i shrnutí, kde uvádím připomínky k programu RFEM a návrhy na možný samostatný modul pro návrh membránových konstrukcí.

Cílem této práce je shrnutí poznatků o metodách návrhu tvaru membránových konstrukcí a analýza vybraných tvarů vhodných pro membránové konstrukce. Diplomová práce by měla být přínosem pro větší porozumění těmto konstrukcím a umožnění jejich širšího uplatnění ve stavebním průmyslu.

2. FORM-FINDING

Form-finding je anglické pojmenování procesu návrhu tvaru membránových konstrukcí. Membránové konstrukce se chovají geometricky nelineárně a to díky své extrémní štíhlosti. Je proto potřeba k nim přistupovat jinak než ke klasickým stavebním konstrukcím. Vzhledem k membránové teorii tyto konstrukce pracují správně jen pokud jsou taženy (v případě, že dojde k tlaku, vzniká na membráně tzv. vrásnění). Cílem návrhu je tedy najít takový tvar, aby hlavní zatížení bylo přenášeno pouze tahovými silami. Abychom dosáhli takového stavu, je potřeba navrhnout konstrukci pomocí některé z metod Form-findingu.

V České republice je oblast hledání vhodného tvaru membránových konstrukcí dosud nerozšířenou mezi odbornou veřejností. Proto se není čemu divit, že literatura dostupná k tomuto tématu se vyskytuje v našich knihovnách velice sporadicky, až dalo by se říci téměř vůbec.

2.1 Historický vývoj

Membránové konstrukce jsou již od nepaměti používány k ochraně proti nepřízní počasí. Ať už to byly tee-pee amerických indiánů, prosté přístřešky lidí z doby kamenné, stany pouštních beduínů nebo cirkusová šapitó. Tvary pro tyto konstrukce vycházely z předchozích zkušeností a stříhů předávaných po generace.

K opravdové analýze vhodného tvaru bylo v minulosti využíváno především fyzického modelování pomocí velice pružných materiálů. Tento přístup se obecně doporučuje provádět i dnes současně s numerickou analýzou pomocí různých softwarových řešení. Lze tak velice jednoduše odhalit případné problémy, které nemusí být na první pohled zřejmé.

2.2 Metody pro hledání tvaru

Tyto metody vychází z teorie minimálního povrchu a teorie mýdlových bublin.

2.2.1 Metoda Force Density

Tuto metodu poprvé publikovali H. J. Schek a K. Linkwitz roku 1973. V inženýrské praxi je hojně využívána k hledání rovnovážného tvaru konstrukce, která sestává z lanové sítě. Zatímco tvarová analýza membránových konstrukcí je geometricky nelineární, FDM linearizuje rovnice pro úpravu tvaru analyticky s využitím poměru hustoty síly pro každý z lanových prvků

$$q = F/L$$

F...síla v prvku

L...délka lanového prvku

2.2.2 Metoda Dynamické relaxace [1]

Základním principem metody dynamické relaxace je využití simulace virtuálního rozkmitání membrány z jejího počátečního stavu. Prvním krokem tedy je diskretizace počáteční plochy, kde je hmotnost konstrukce koncentrována do předem definovaných bodů, kterými je pokryt její na povrch. Tím vznikne síť nehmotných lan s hmotnými uzly. Systém těchto hmotných bodů je pak excitován virtuální silou a vibruje okolo jedné rovnovážné polohy s ohledem na aplikované zatížení. Provedeme-li časovou analýzu tohoto systému vystaveného virtuální dynamické excitaci společně s použitím kinematického tlumení, systém po určité době dosáhne rovnovážného stavu. Tato časová analýza pseudo-dynamického procesu vychází ze základní rovnice dynamiky

$$P_{ji} = M_{ji} \ddot{\delta}_{ji} + C_{ji} \dot{\delta}_{ji} + [\Sigma K \delta]_{ji}, \quad (2.1)$$

kde pro diskretizovaný model index ji odpovídá j -tému bodu v i -tém směru, δ_{ji} , $\dot{\delta}_{ji}$ jsou vektory bodového zrychlení a rychlosti. P_{ji} je externí zatížení, $[\Sigma K \delta]_{ji}$ je suma vektorů vnitřních sil (kde K je tuhost a δ přesuny), M_{ji} je hmotnost jednotlivých bodů a C_{ji} koeficient viskózního tlumení.

Zavedeme bodové residuální síly R_{ji} , které vznikají rozdílem mezi vektorem externího zatížení a vnitřních sil.

$$R_{ji} = P_{ji} - [\Sigma K \delta]_{ji} \quad (2.2)$$

Z rovnic (2.1) a (2.2) vyplývá, že

$$R_{ji} = M_{ji} \ddot{\delta}_{ji} + C_{ji} \dot{\delta}_{ji} \quad (2.3)$$

Pokud tuto diferenciální rovnici zapíšeme ve formě diferenčních rovnic, pak následné hodnoty zrychlení a rychlostí v čase $n\Delta t$ (kde n označuje číslo iteračního kroku a Δt přírůstek času) lze zapsat jejich hodnotami z předešlé iterace.

$$R_{ji}^n = P_{ji} - [\Sigma K \delta]_{ji}^{n-1} = M_{ji} \ddot{\delta}_{ji}^n + C_{ji} \dot{\delta}_{ji}^{n-1} \quad (2.4)$$

Z rovnice (2.4) je odvozeno zrychlení

$$\ddot{\delta}_{ji}^n = \frac{R_{ji}^n - C_{ji} \dot{\delta}_{ji}^{n-1}}{M_{ji}}, \quad (2.5)$$

rychlosti

$$\dot{\delta}_{ji}^n = \dot{\delta}_{ji}^{n-1} + \ddot{\delta}_{ji}^n \Delta t, \quad (2.6)$$

a přesuny

$$\delta_{ji}^n = \delta_{ji}^{n-1} + \dot{\delta}_{ji}^n \Delta t. \quad (2.7)$$

Celý iterační proces je pak složen z opakování rovnice (2.4) s následným vyhodnocením rovnic (2.5) - (2.7).

2.2.2.1 Kinetické tlumení

Průběh řešení je možné zjednodušit pomocí užití kinetického tlumení. To znamená použití nulového tlumení C , zatímco je vibrace konstrukce kontrolována pomocí zjišťování celkové kinetické energie. Jakmile kinetická energie systému dosáhne svého maxima, je iterační proces zastaven. Následně jsou iterace restartovány z aktuální konfigurace s nulovou počáteční rychlostí. S postupem času se vrcholy kinetické energie zmenšují, jak se nevyrovnané síly a rychlosti zmenšují, až se konstrukce postupně ustálí ve stavu statické rovnováhy.

Konvergence statického řešení je dosažena díky kumulačnímu efektu postupných redukcí kinetické energie. Tato metoda závisí na pozorování, že při jednoduchém harmonickém pohybu je maximální kinetická energie dosažena v okamžiku, kdy je minimální potenciální energie. Tento princip nejlépe ilustruje pohyb kyvadla.

Pokud nastavíme viskozní tlumení rovno nule, dostaneme z rovnice (2.5)

$$\ddot{\delta}_{ji}^n = \frac{R_{ji}^n}{M_{ji}^n}, \quad (2.8)$$

zatímco výrazy pro rychlosti (2.6) a přesunutí (2.7) zůstanou nezměněny.

V iteračním procesu pak dojde pouze k použití rovnice (2.8) namísto (2.5) oproti algoritmu bez kinetického tlumení.

2.2.3 Metoda Updated reference strategy [2]

Základní rovnice, která popisuje tuto metodu hledání tvaru, je založena na principu virtuální práce vnitřních sil.

$$|\delta w = t \int_a \sigma : \frac{\partial(\delta u)}{\partial x} da = t \int_a \sigma : \delta u_{,x} da = 0 \quad (2.9)$$

Rovnice (2.9) vyjadřuje závislost, kdy při rovnovážném stavu bude pole napětí nulové. σ je předepsaný Cauchyho tenzor napětí, který působí na povrch v klidu, $\delta u_{,x}$ je derivace virtuálního přemístění respektující aktuální plochu. Tloušťka membrány je ve vzorci definována jako t .

Bohužel přímé numerické řešení rovnovážné rovnice (2.9) není možné. Matice tohoto diskretizovaného systému by byla singularní. Řešení tohoto problému leží v inverzním přístupu. Hledáme diskretizovaný rovnovážný povrch s určitou topologií diskretizačních parametrů, která vyúsťuje do kontextu MKP jako síť, přesněji řečeno jako pozice bodů. Je však dokázáno, že body mohou být libovolně přemísťovány po rovnovážném povrchu, aniž by to bylo v rozporu s podmínkami rovnováhy (resp. zachovávají platnou diskretizaci hledaného povrchu). Říká se jim tzv. „plovoucí síť“. Ovšem neexistuje žádný určitý typ sítě, který by způsobil singularitu a zabránil by přímému řešení. Proto byla vyvinuta metoda „Strategie aktualizované referenční plochy“.

Tato metoda je velice silnou díky tomu, že je přímo odvozená z nelineární mechaniky kontinua. Její myšlenkou je nahradit původní singulární úlohu relativní, dobře postavenou úlohou. Namísto předpokladu, že máme dán Cauchyho tenzor napětí σ neznámého rovnovážného povrchu, stanovíme tenzor druhého napětí Piola-Kirchhoff \mathbf{S} , který odpovídá libovolné počáteční geometrii plochy. Tato dvě rozdílná napětí jsou propojena deformačním gradientem \mathbf{F} , který je měřítkem vzniklého přesunu mezi referenční a aktuální geometrií.

$$\mathbf{S} = \det \mathbf{F} \mathbf{F}^{-1} \cdot \sigma \cdot \mathbf{F}^{-T} \quad (2.10)$$

Je potřeba poznamenat, že pouze Cauchyho tenzor napětí představuje reálné fyzikální napětí, které působí na výsledném povrchu. Tenzor druhého napětí Piola-Kirchhoff odpovídá nedeformované geometrii a je to pouze veličina, která je použita za účelem výpočtu. Pokud je aktuální konfigurace identická s referenční, jsou tyto míry také identické.

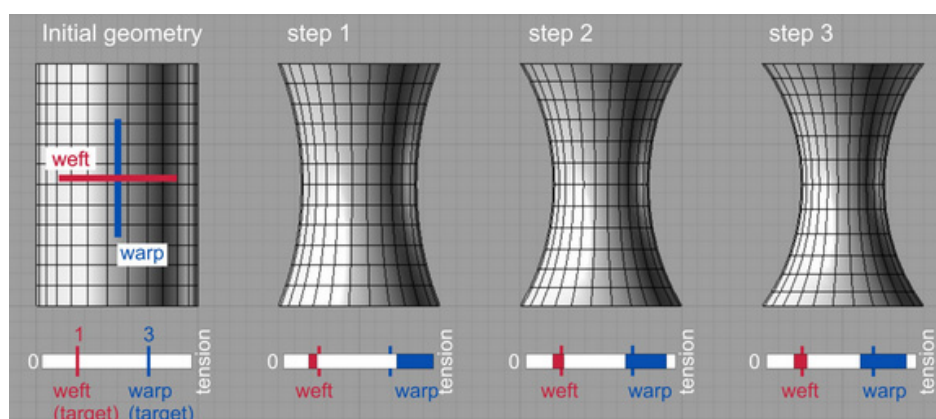
Derivace začíná rovnicí (2.9), ve které je Cauchyho tenzor napětí nahrazen tenzorem druhého napětí Piola-Kirchhoff

$$\delta w = t \int_A \left(\mathbf{F} \cdot \left(\det \mathbf{F} \mathbf{F}^{-1} \cdot \sigma \cdot \mathbf{F}^{-T} \right) \right) : \delta \mathbf{F} dA = t \int_A \left(\mathbf{F} \cdot \mathbf{S} \right) : \delta \mathbf{F} dA \quad (2.11)$$

Pomocí této rovnice jsme schopni vypočítat rovnovážný povrch pro daný stav napětí dle daného tenzoru druhého Piola-Kirchhoff napětí. Výsledné Cauchyho napětí se však bude od hledaného stavu napětí lišit. Proto je nutné rozšířit daný algoritmus o další iterační smyčku. Získaný rovnovážný povrch (který je bližší výslednému povrchu) tedy označíme jako referenční pro následující iterační krok. Toto představuje podstatu návrhu celého procesu, který je proto nazýván „Updated Reference Strategy“. Po aktualizaci referenční plochy opět specifikujeme tenzor druhého napětí Piola-Kirchhoff v závislosti na hledaném poli napětí, abychom dostali dobře definovanou úlohu a abychom byli schopni vypočítat novou aktuální konfiguraci. Současně se zmenšujícím se odpovídajícím přemístěním mezi aktuální a referenční konfigurací, se zmenšuje i rozdíl mezi druhým napětím Piola-Kirchhoff a Cauchyho napětím. Tímto pak výpočet konverguje k úloze, která se už nebude měnit.

Tato extrémě efektivní metoda umožňuje použít jakýkoliv stav napětí s nebo bez vnějšího zatížení. Navíc může být použita s jakoukoliv diskretizací a použita stejným způsobem.

Zvláštní důraz je pak kladen na výpočet, pokud ve směrech waft a warp má být předpětí s různou hodnotou a tudíž je předepsáno anisotropní rozdělení napětí. Matematicky je dokázáno, že pouze plochy s nulovou Gaussovou křivostí (jako rovinné plochy nebo válce) mají konstantní anisotropní pole napětí. Pro všechny ostatní plochy s dvojitou křivostí (což je normální u většiny membránových konstrukcí) platí, že vznikají mírné odchylky od výsledného napětí a musí být přijaty, protože neexistuje žádné přesnější matematické řešení tohoto problému. Jednotlivá řešení pro všechny kroky představují fyzikálně stabilní řešení. Uživatel tak musí upravit pouze počet provedených kroků hledání tvaru tak, aby řešení splňovalo požadavky na geometrii a přesnost vyplývajících napětí.



Obr. 1: Odchylky napětí na katenoidu s anisotropním předpětím [2]

2.2.4 Nelineární analýza [3]

2.2.4.1 Updated Lagrangian

Pro použití této metody je potřeba znát základní rovnice z oblasti fyziky a rovnice popisující vlastnosti standardního Boltzmanova continua.

Zákon o zachování hmoty

Jedná se o zákon popisující změnu hustoty tělesa v závislosti na deformaci.

$$\rho(X) = \frac{\rho_0(X)}{\det(F)} = \frac{\rho_0(X)}{J}$$

ρ_0 ...původní hustota (na referenční konfiguraci)

ρ ...běžná hustota (na deformovaném tělese)

Zákon o zachování hybnosti

Zákon o zachování lineární hybnosti

$$\nabla \cdot \sigma + \rho \cdot b = 0$$

$\rho \cdot b$...vektor objemových sil

b ...vektor gravitačního zrychlení

Zákon o zachování točivosti (úhlové hybnosti)

Vyjádření momentové podmínky rovnováhy při zanedbání setrvačných sil. Z rovnice je jasná i symetrie tenzoru napjatosti σ .

$$\sigma = \sigma^T$$

Zákon o zachování energie

Tento zákon v mechanice těles popisuje, že rychlost změny celkové energie tělesa je rovna součtu rychlosti práce vnějších sil (výkonu zatížení), tepelného toku a zdroje energie.

Konstitutivní rovnice

Tato rovnice popisuje vztah mezi napětím a deformací na deformovaném tělese.

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{e}, \boldsymbol{\sigma}, \dots)$$

Geometrické rovnice (míra deformace)

$$\mathbf{e} = \frac{1}{2}(\mathbf{I} - \mathbf{F}^{-T} \cdot \mathbf{F}^{-1})$$

V této metodě se používá Euler – Almansiho tenzor deformace definovaný na deformovaném tělese.

Samotná diskretizace MKP pro formulaci na běžné konfiguraci (updated Lagrangian), jak již bylo řečeno, je definována na deformovaném tělese Ω . Vztah mezi virtuálním přírůstkem Euler – Almansiho tenzoru deformace zapsaným ve Voigtově notaci $\delta \{e\}$ a virtuálním přírůstkem vektoru parametrů deformace $\delta \mathbf{d}$ vyjadřuje rovnice:

$$\delta \{e\} = \mathbf{B} \cdot \delta \mathbf{d}$$

Při této definici může být vektor vnitřních uzlových sil počítán podle vzorce:

$$\mathbf{f}^{\text{int}} = \int_{\Omega} \mathbf{B}^T \{\boldsymbol{\sigma}\} d\Omega$$

Tečná matice tuhosti

Tečná matice tuhosti charakterizuje běžnou tuhost v daném okamžiku. Respektuje tedy změnu geometrie, tečnou tuhost materiálu a vliv napjatosti. Takto lze tuto matici tuhosti v běžné konfiguraci \mathbf{d} definovat jako:

$$\mathbf{K}_T = \mathbf{K}_M + \mathbf{K}_G$$

\mathbf{K}_M ...tečná materiálová matice tuhosti

\mathbf{K}_G ...tečná geometrická matice tuhosti

2.2.4.2 Newton – Raphsonova iterační metoda

Tato metoda je založena na hledání řešení, při kterém jsou nevyvážené síly $r(d)$ nulové.

Po provedení rozvoje $r(d)$ do Taylorovy řady kolem známého řešení $d^{(i-1)}$, kde δd je přírůstek $\delta d^{(i)} = d^{(i)} - d^{(i-1)}$ a pokud zanedbáme členy druhého a vyšších řádů, rovnice bude vypadat následovně:

$$\delta d^{(i)} = -\left(K_T(d^{(i-1)})\right)^{-1} \cdot r(d^{(i-1)}) = \left(K_T(d^{(i-1)})\right)^{-1} \cdot \left(f - K(d^{(i-1)}) \cdot d^{(i-1)}\right)$$

K_T ...tangenta (sklon) čáry $r(d)$ v $d^{(i-1)}$ – při řešení úloh deformační variantou MKP se tato tangenta nazývá tečnou maticí tuhosti

Pokud tato procedura konverguje, nevyvážená síla $r(d)$ postupně klesá k nule.

Celkové řešení pro přírůstek d v i – té iteraci je následující:

$$d^{(i)} = d^{(i-1)} + \delta d^{(i)}$$

Newton-Raphsonovu proceduru lze pro soustavu nelineárních rovnic zapsat takto:

$$\delta d = K_T^{-1} r$$

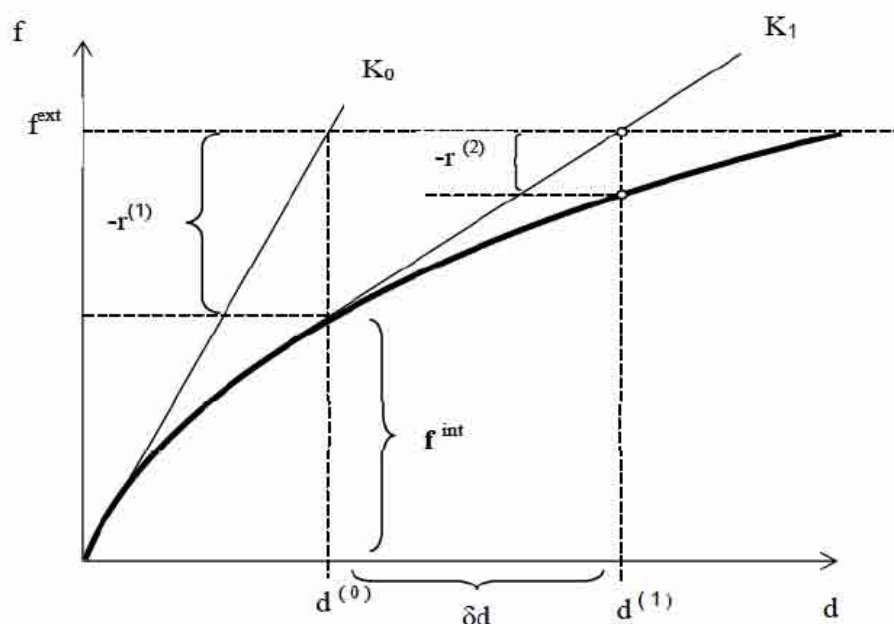
K_T ...tečná matice

$$K_T^{(i)} = \left. \frac{\partial \mathbf{r}}{\partial \mathbf{d}} \right|_{\mathbf{d}^{(i-1)}}$$

\mathbf{r} je vektor nevyváženého zatížení

$$\mathbf{r} = \mathbf{f}^{\text{int}} - \mathbf{f}^{\text{ext}}$$

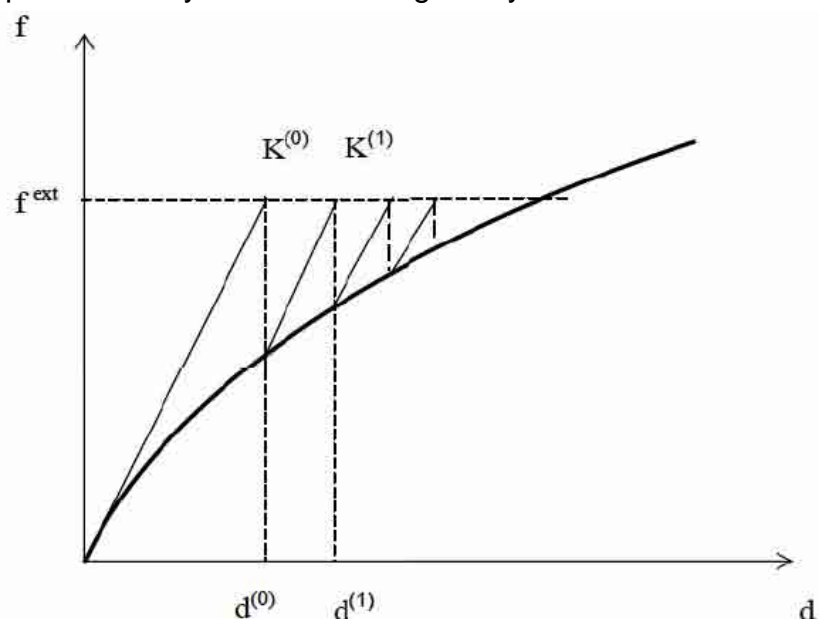
Princip Newton – Raphsonovy metody lze znázornit i graficky:



Obr. 2 Princip Newton – Raphsonovy metody

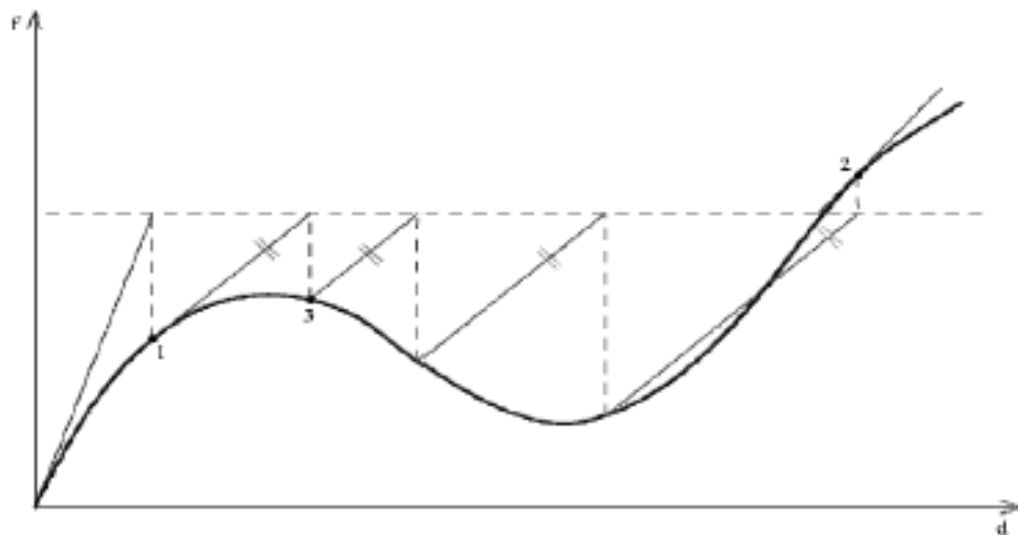
Pro samotný výpočet je však často výhodnější použít modifikovanou Newton – Raphsonovu metodu, která nám dává možnost ponechat levou stranu soustavy rovnic beze změny a měnit pouze pravou stranu. Toto nám umožní, narozdíl od normální Newton – Raphsonovy metody, provádět dekompozici matice při řešení Gaussovou nebo Choleského metodou jen jednou a tím i při vyšším počtu nutných iterací zajistí rychlejší výpočet.

Princip této metody lze znázornit i graficky:



Obr. 3 Princip modifikované Newton – Raphsonovy metody

Obě tyto metody (modifikovaná a normální) se dají výhodně kombinovat a to jednak kvůli úspoře času při výpočtu a jednak k řešení úloh, pro které by použití nemodifikované Newton – Raphsonovy metody selhalo.

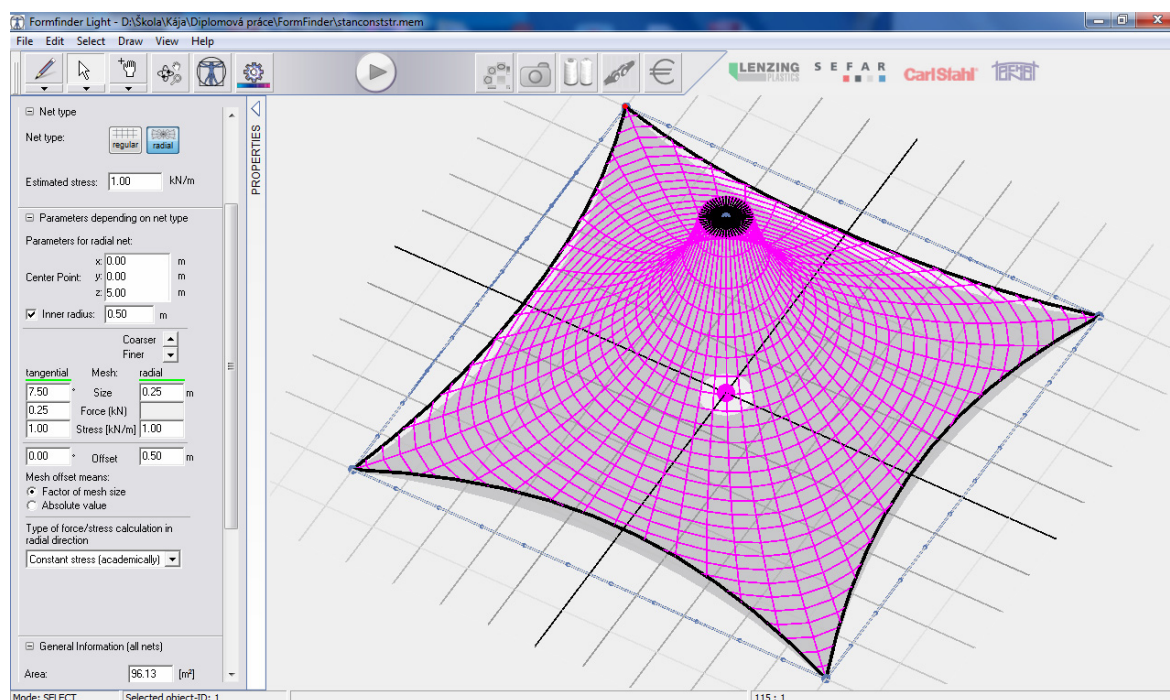


Obr. 4 Kombinace Newton-Raphsonovy a modifikované Newton-Raphsonovy metody

2.3 Programy na hledání vhodného tvaru

2.3.1 Form Finder

Tento program byl vyvinut přímo pro návrh tvarů membránových konstrukcí na základě letitého výzkumu v této oblasti. Za programem stojí firma pana Dipl.-Ing. Dr. Techn. Robert Wehdorn-Roithmayr.



Metoda výpočtu vhodných tvarů není uvedena. Veškerá dokumentace je k dispozici až ve verzi Profi, která je však již placená (cena neuvedena).

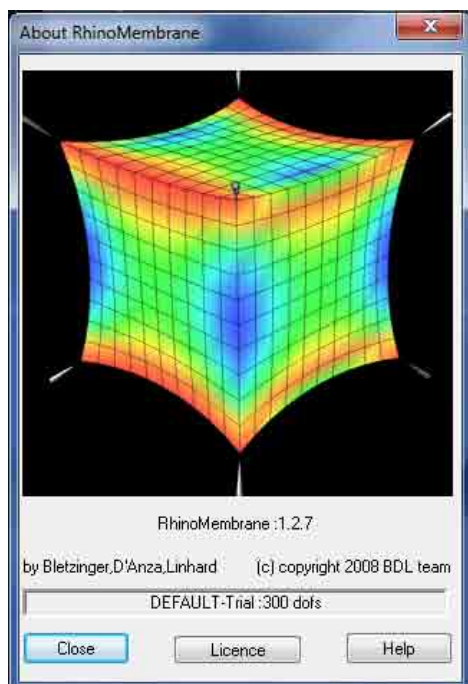
2.3.2 RhinoMembrane

Jako rozšíření programu Rhinoceros je možné dokoupit modul RhinoMembrane, který využívá metodu Updated Reference Strategy popsanou výše. Tato metoda využívá namísto předepsání materiálového pole \mathbf{X} (i.e. množství a umístění hmoty v prostoru, který tvoří konstrukci) přípustné pole napětí \mathbf{S} , pro které je pak určen rovnovážný tvar.

Uživatel namodeluje pomocí NURBS ploch v programu Rhino výchozí tvar konstrukce, který respektuje okrajové podmínky. Následně tuto konstrukci pokryje sítí. V případě spojených ploch je třeba zajistit, aby hraniční body sítí na spojích

jednotlivých ploch byly sjednocené. Následně musí specifikovat distribuci předpětí v membráně a předpínacích kabelech jako vstupní data pro proces hledání tvaru.

Pro pneumatické stavby jako třeba nafukovací zastřešení hal je dokonce možné aplikovat tlak p v rámci hledání tvaru. V neposlední řadě je také potřeba definovat dostatečné okrajové podmínky, abychom dostali jedinečné řešení rovnovážného tvaru.



V trial verzi, která je zdarma, je možné modelovat konstrukci s omezením počtu bodů sítě na 100. Studentská verze s neomezeným počtem bodů je pak za 450,76 EUR (kompletní balení).

Proto bohužel nebylo možné analyzovat ani jeden tvar. Pokud je však tento program k dispozici jistě je silným nástrojem pro hledání vhodného tvaru membránových konstrukcí.

2.3.3 Skript na relaxaci

Program Rhinoceros nabízí v oblasti vytváření skriptů široké možnosti. Proto pan David Rutten vytvořil během svého pobytu na Prat-SA, Toulouse, ve Francii, dva skripty pro relaxaci ploch. Oba skripty jsou založeny na Force Density Method a redukci vibrací. Liší se ovšem použitím.

První se jmenuje MeshRelaxation.rvb a je určen pro relaxaci sítí.

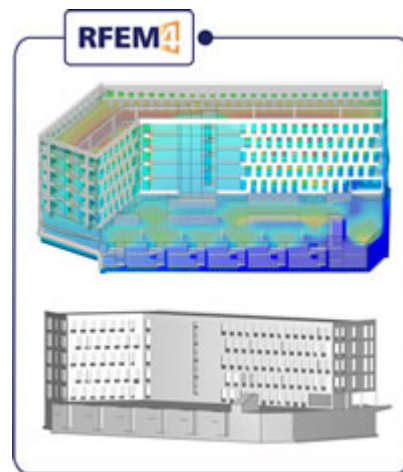
Druhý má název NurbsRelaxation.rvb a s jeho pomocí je možné hledat minimální tvar NURBS ploch. Tento skript má ovšem omezení, že umožňuje relaxaci pouze jedné NURBS plochy, která může být ovšem rotační či cyklická. Tzn. nesmí se jednat o spojené NURBS plochy. Na rozdíl ale od prvního skriptu dokáže použít různé napětí pro dva na sebe kolmé směry (weft a warp).

Tyto skripty jsou volně dostupné na internetu pro nekomerční použití.

2.3.4 RFEM

RFEM 4 je program pro 3D analýzu metodou konečných prvků (MKP), vhodný pro náročné inženýry – statiky, který splňuje požadavky moderního stavebního inženýrství a odráží nejnovější trendy v oboru.

RFEM je založen na modulárním softwarovém systému: základní program RFEM určuje vnitřní síly, deformace a podporové reakce pro konstrukce složené z ploch a prutů. Pro následná posouzení jsou k dispozici přídavné moduly, které odpovídají specifickým požadavkům na materiály a normy. Díky této modulární koncepci lze sestavit sadu programů, která bude nejlépe vyhovovat individuálním přáním uživatele.



V současné době je k dispozici RFEM 4, která umožňuje nelineární analýzu III. stupně s využitím Newton-Raphsonovy iterační metody. V dohledné době se však připravuje RFEM 5, kde bude možné k výpočtu použít jak Newton-Raphsonovu metodu, tak Picardovu či jejich kombinaci. Navíc umožní výpočet i pomocí dynamické relaxace.

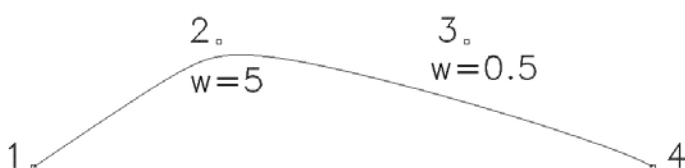
2.4 NURBS křivky a plochy

Zkratka NURBS bývá mnohdy označována jako Nobody Understands Rational B-Spline. Ve skutečnosti se jedná o zkratku názvu Non-uniform rational B-spline.

B-spline jsou aproximační křivky a plochy dané řídicími body a spjaté s B-spline funkcemi. Tyto funkce jsou zadány rekurentním vztahem a jejich tvar závisí na uzlovém vektoru.

Non-uniform znamená, že rozdíl sousedních čísel v uzlovém vektoru není konstantní. Tj. křivka nemá plynulý průběh křivosti, což umožňuje lepší možnosti při úpravě tvaru.

Přiřazením váhy k jednotlivým bodům (tzn. každému bodu je přiřazeno číslo, které udává, jak moc bude křivka k danému bodu přitahována), pak do názvu přibude pojem *rational*. Ve většině případů je přiřazena všem bodům hodnota váhy rovna 1. V tomto případě se pak jedná o non-rational křivku. Používat hodnotu $w=1$ se doporučuje i v případech, když je nutné následně křivku nebo plochu převést do jiného programu. Pokud nastavíme hodnotu váhy v intervalu $(0;1)$, znamená to, že vliv daného bodu na křivku je malý a křivka bude vzdálená od bodu víc. V případě, že ji nastavíme větší než jedna, zvyšuje se síla, kterou je křivka přitahována k danému bodu. Geometricky je výpočet plochy s váhami řešen rozšířením 3D prostoru o další dimenzi.



Obr. 5 Vliv váhy jednotlivých bodů na tvar křivky

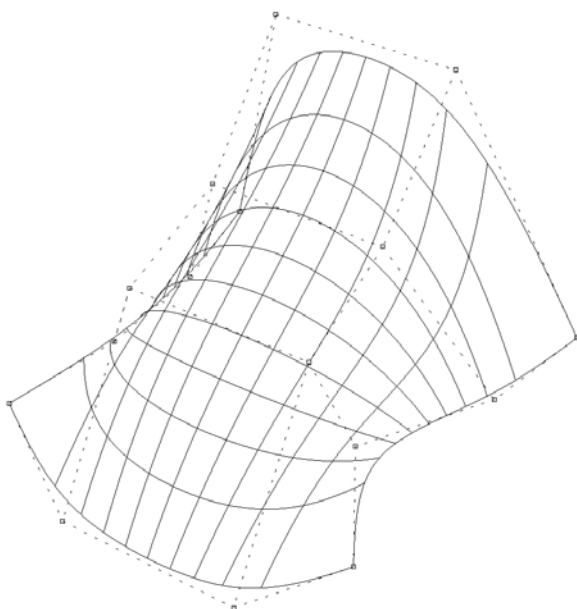
Váha je přiřazena jako čtvrtá souřadnice bodu a do výpočtu je použita stejným způsobem jako ostatní souřadnice. Po skončení výpočtu se první tři vypočtené hodnoty podělí danou vahou. Matematický zápis tohoto výpočtu je:

$$C(t) = \frac{\sum_{i=0}^m w_i P_i N_i^n(t)}{\sum_{i=0}^m w_i N_i^n(t)} \quad (2.12)$$

NURBS křivky a plochy se využívají v CAD systémech díky jejich výborným modelačním schopnostem. Změnou polohy řídicího bodu, hodnoty váhy či tvarem uzlového vektoru lze měnit lokálně tvar křivky nebo plochy.

Velkou výhodou křivek NURBS je jejich schopnost přesně vyjádřit kuželosečky. Proto jimi lze jednoduše konstruovat základní tělesa (kužel, válec, anuloid, atd.). Obecná NURBS plocha pak má poměrně složitější matematický zápis:

$$C(u, v) = \frac{\sum_{i=0}^q \sum_{j=0}^r w_{ij} P_{ij} N_i^m(u) N_j^n(v)}{\sum_{i=0}^q \sum_{j=0}^r N_i^m(u) N_j^n(v)} \quad (2.13)$$



Obr. 6 NURBS plocha s řídicími body

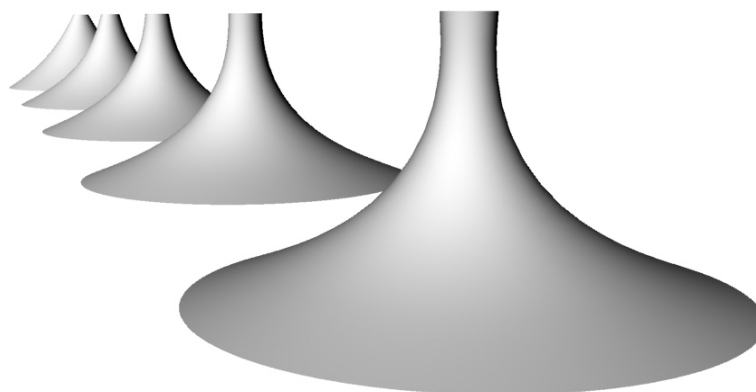
3. ANALÝZA VYBRANÝCH TVARŮ V PROGRAMU RFEM

Ve většině případů jsou membránové konstrukce navrhovány pro specifické umístění podpor dle prostoru, který má být zastřešen. Proto nelze najít jedno vhodné řešení pro jakékoliv okrajové podmínky. Pro názornost jsem proto zvolil dva způsoby uchycení membránových konstrukcí. V prvním případě se jedná o rotační plochy, kdy vnitřní i vnější hrana má po celé délce zabráněno v pohybu pomocí liniových podpor. V druhém případě se jedná o membránu čtvercového půdorysu, kdy na vnějším obvodu je v rozích uchycena pomocí čtyř bodových podpor a mezi nimi předpjatým kabelem. Na vnitřním obvodu je opět liniová podpora. K modelování membránových konstrukcí čtvercového půdorysu jsem jako nejvhodnější zvolil NURBS plochy, které umožňují vytvoření modelů s hladkou plochou. Všechny tvary byly zatíženy vlastní tíhou textilie, předepnutím a následným zatížením sněhem s referenční hodnotou 1 kN/m^2 pro celou plochu membrány. Ve výpočtu zatěžovacích kombinací bylo přihlédnuto i k bezpečnostním součinitelům platným pro jednotlivé typy zatížení dle platného eurokódu.

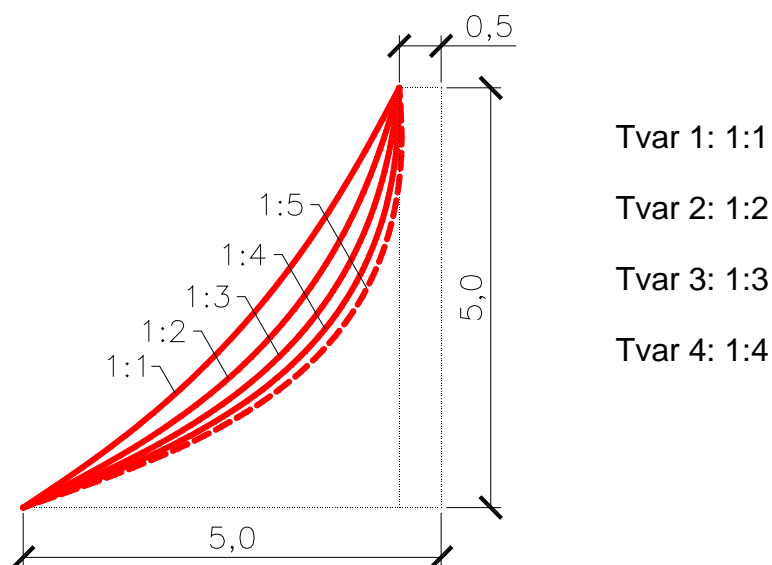
3.1 Tvary vytvořené pomocí Rhino skriptů

3.1.1 Rotační plocha

Vzhledem k povaze této plochy je možné ji vymodelovat pomocí jedné rotační NURBS plochy a proto byl k nalezení jejího tvaru použit skript NurbsRelaxation. Tento skript umožňuje ovlivnit výsledný tvar pomocí poměru napětí ve dvou směrech, označovaných jako weft a warp.



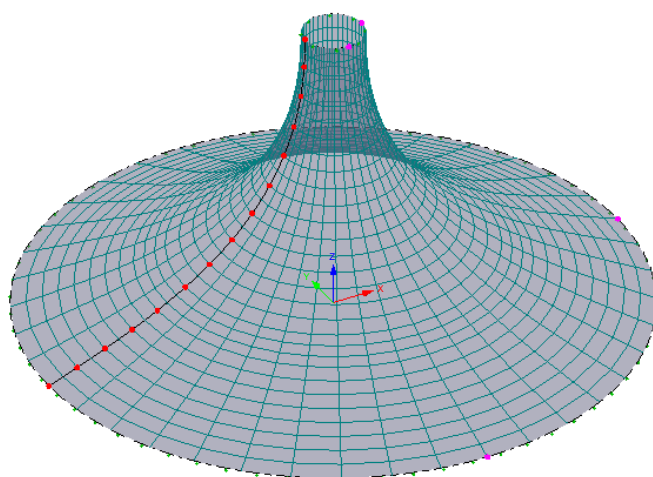
Obr. 7 Perspektiva analyzovaných tvarů



Obr. 8 Řezy rotační plochou s různými poměry weft:warp

Z obr. 8 je zřejmé, že při poměru weft:warp=1:5 již dochází u vrcholu membrány k prohnutí plochy dovnitř, což z estetického hlediska není žádoucí.

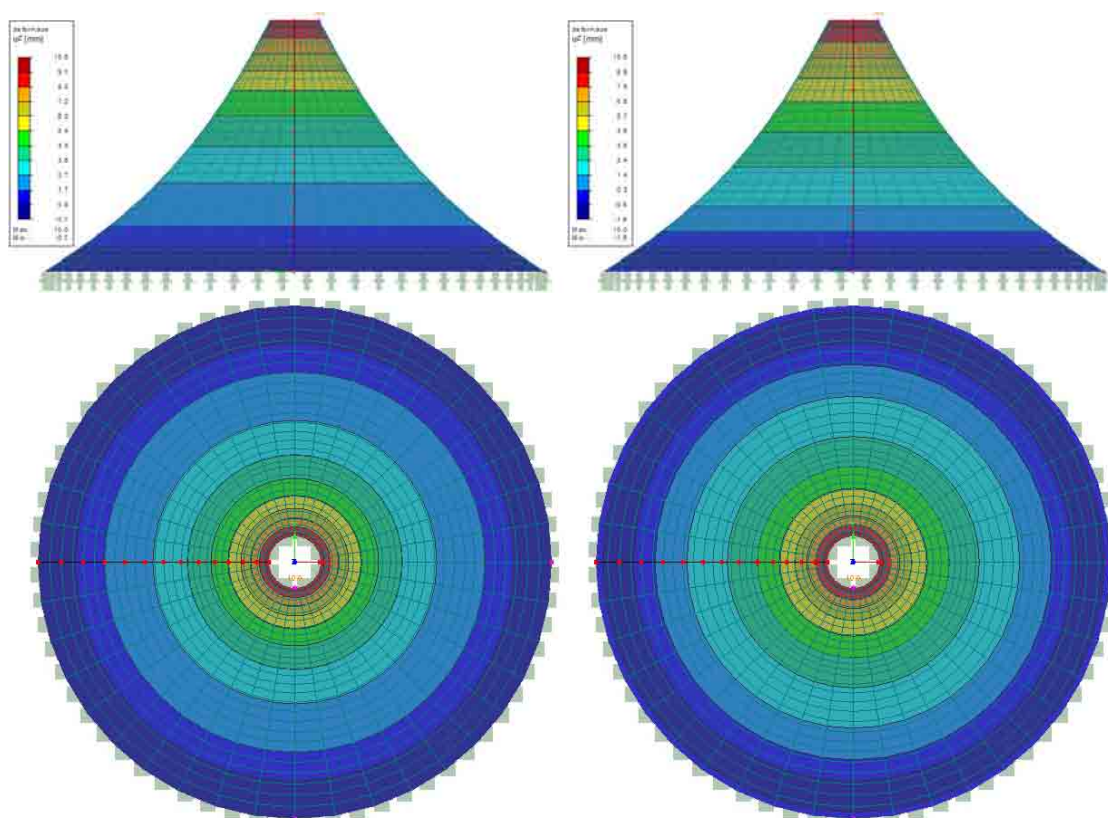
Proto budu analyzovat tvary s poměrem weft:warp do 1:4. Pro přenesení geometrie do programu RFEM byla plocha pokryta sítí a exportována ve formátu *.dxf. Pokrytí sítí bylo provedeno z důvodu chybějící podpory NURBS entit formátem DXF. V programu RFEM byl pak výsledný tvar určen řezem (spline křivkou) a jeho následnou rotací okolo osy z. Předpětí do membrány bylo vneseno vynuceným posunem vnitřní liniové podpory o 10 mm ve směru osy z.



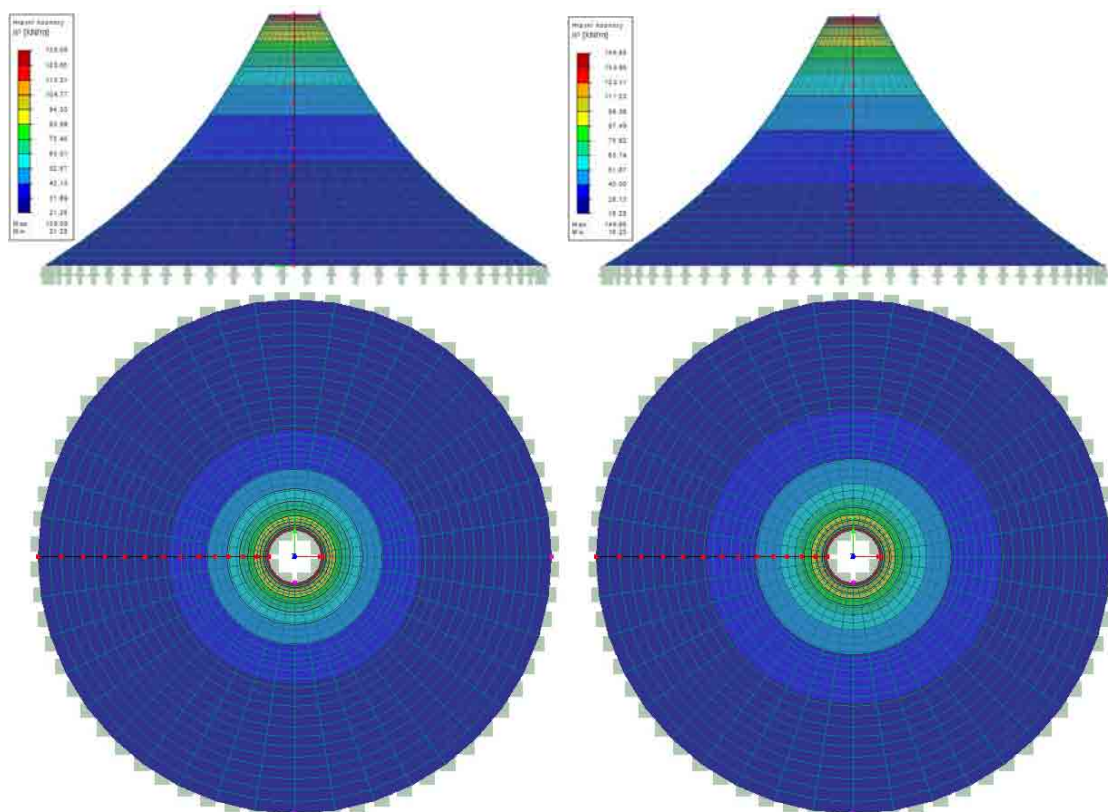
Obr. 9 Příklad tvaru vytvořeného rotací spline křivky kolem osy z

3.1.1.1 Tvar 1

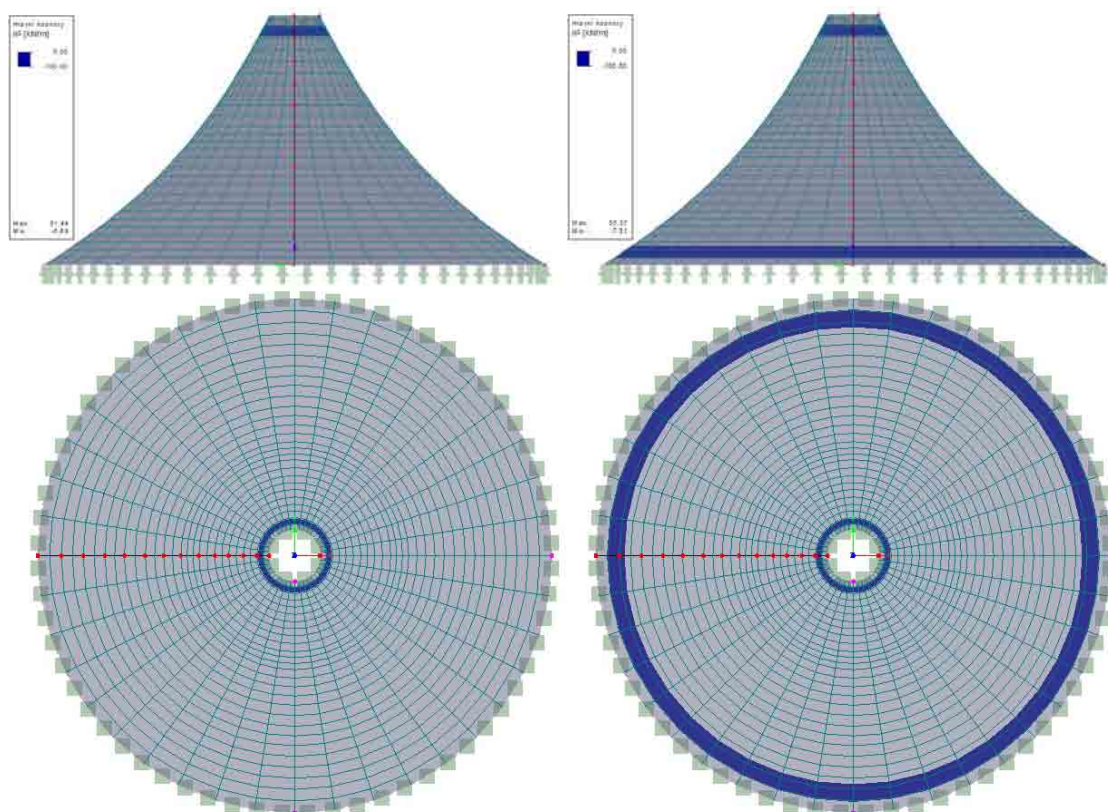
Na prvním tvaru vznikly deformace při stálém zatížení velikosti max. 10 mm, což odpovídá deformaci vnitřní liniové podpory. Při zatížení sněhem zůstala maximální deformace na 10 mm. Hlavní napětí n_1 vzrostlo ze 136.09 kN/m pro stálé zatížení na 146.85 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -6.69 kN/m pro stálé zatížení na -7.31 kN/m pro kombinaci se sněhem. Společně se zatížením sněhem vznikl pás příčných tlaků v dolní části membrány. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 10 Deformace při stálém zatížení a kombinaci stálého se sněhem



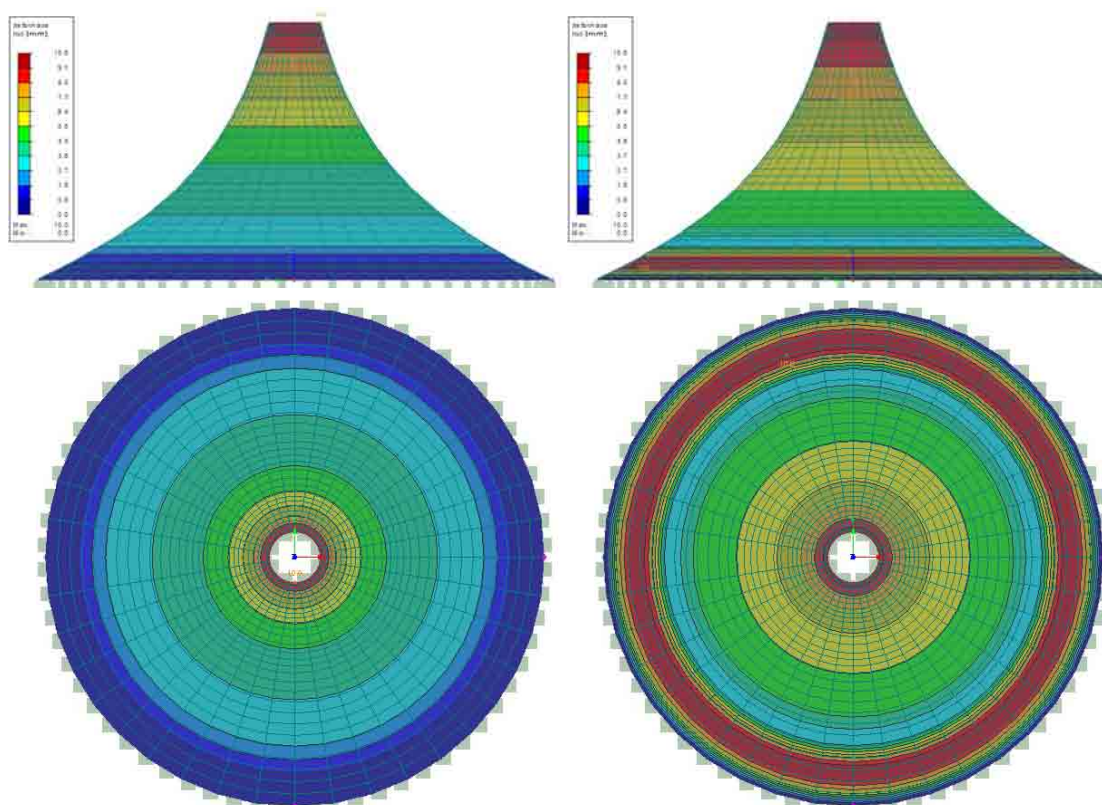
Obr. 11 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



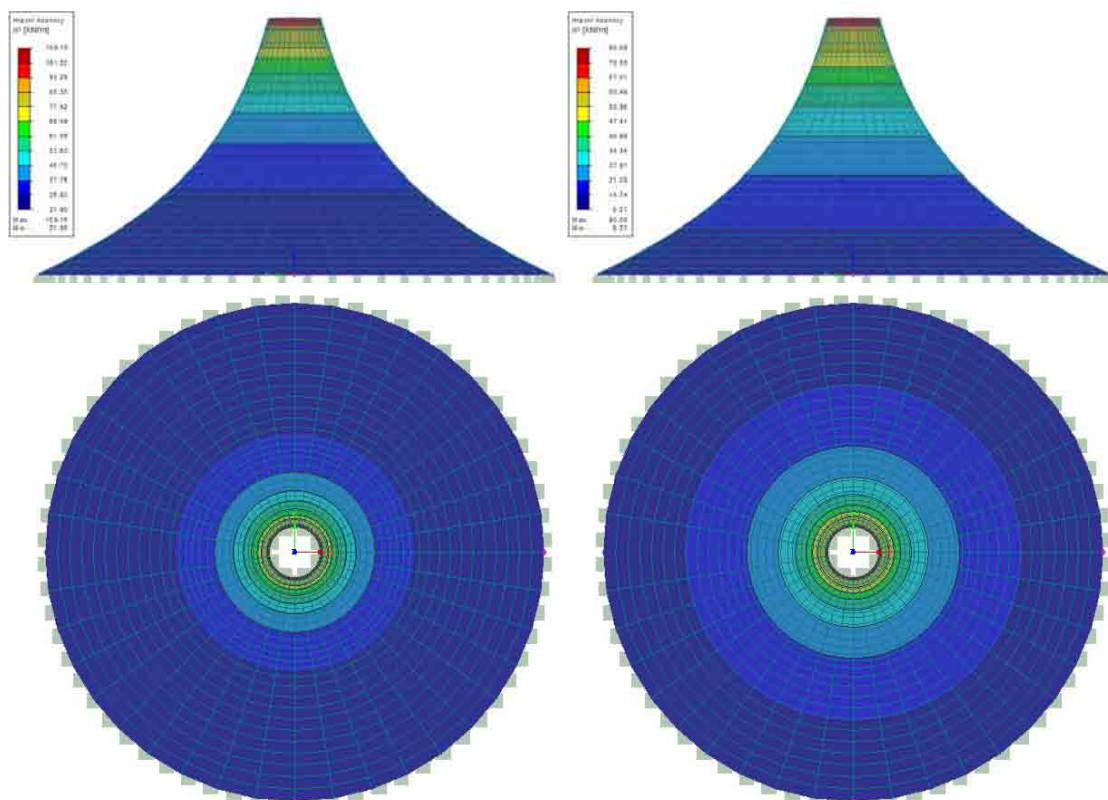
Obr. 12 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.1.2 Tvar 2

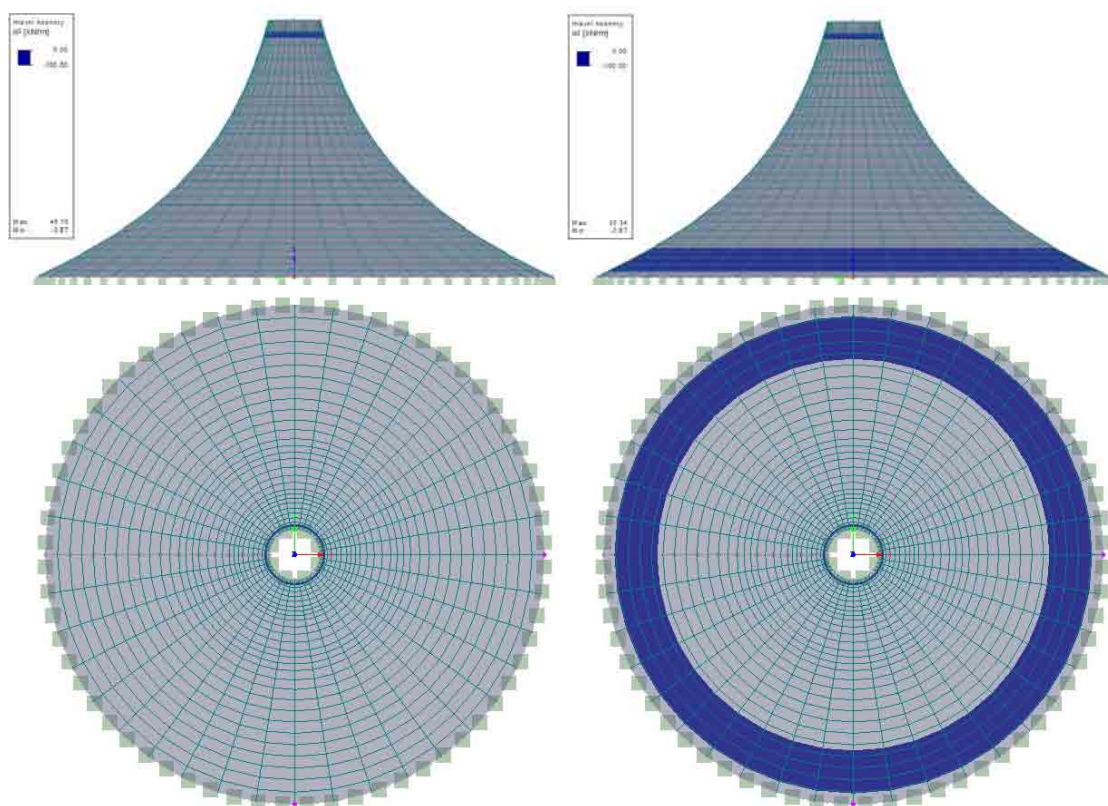
Na druhém tvaru opět vznikly deformace při stálém zatížení velikosti vynuceného posunu podpor, tj. 10 mm. Při zatížení sněhem však došlo ke stejnému přemístění uzlů i v dolní části membrány. Hlavní napětí n_1 kleslo ze 109.15 kN/m pro stálé zatížení na 80.08 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 vzrostly z -3.87 kN/m pro stálé zatížení na -2.97 kN/m pro kombinaci se sněhem. Pás příčných tlaků v dolní části membrány má sice nižší hodnoty než u předchozího tvaru, avšak je rozprostřen na větší plochu. Pás tlaků u vrcholu membrány však svoji plochu zmenšil. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 13 Deformace při stálém zatížení a kombinaci stálého se sněhem



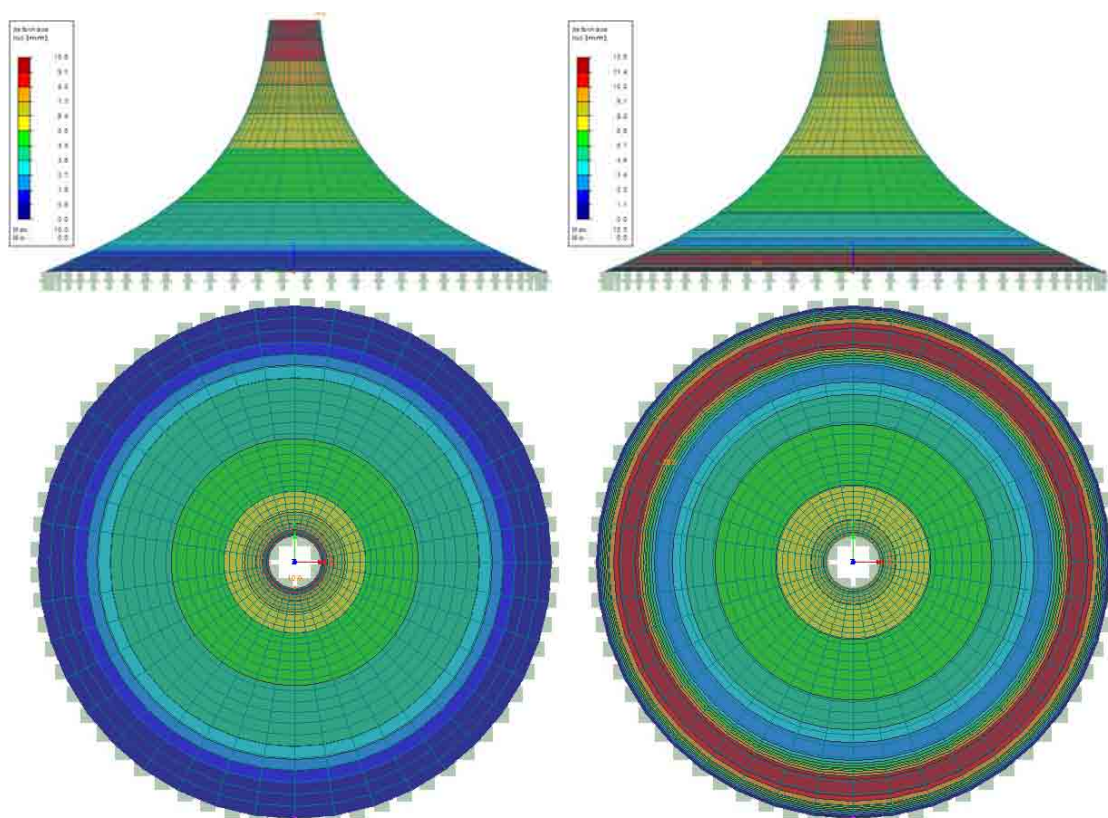
Obr. 14 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



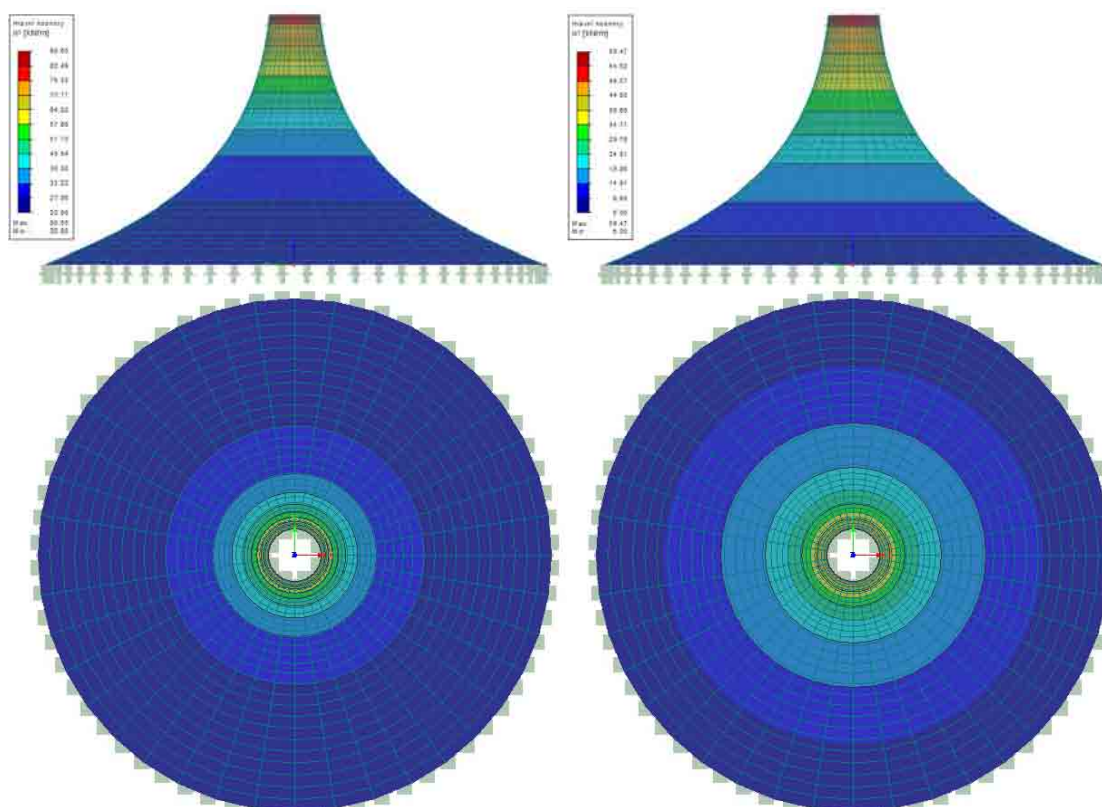
Obr. 15 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.1.3 Tvar 3

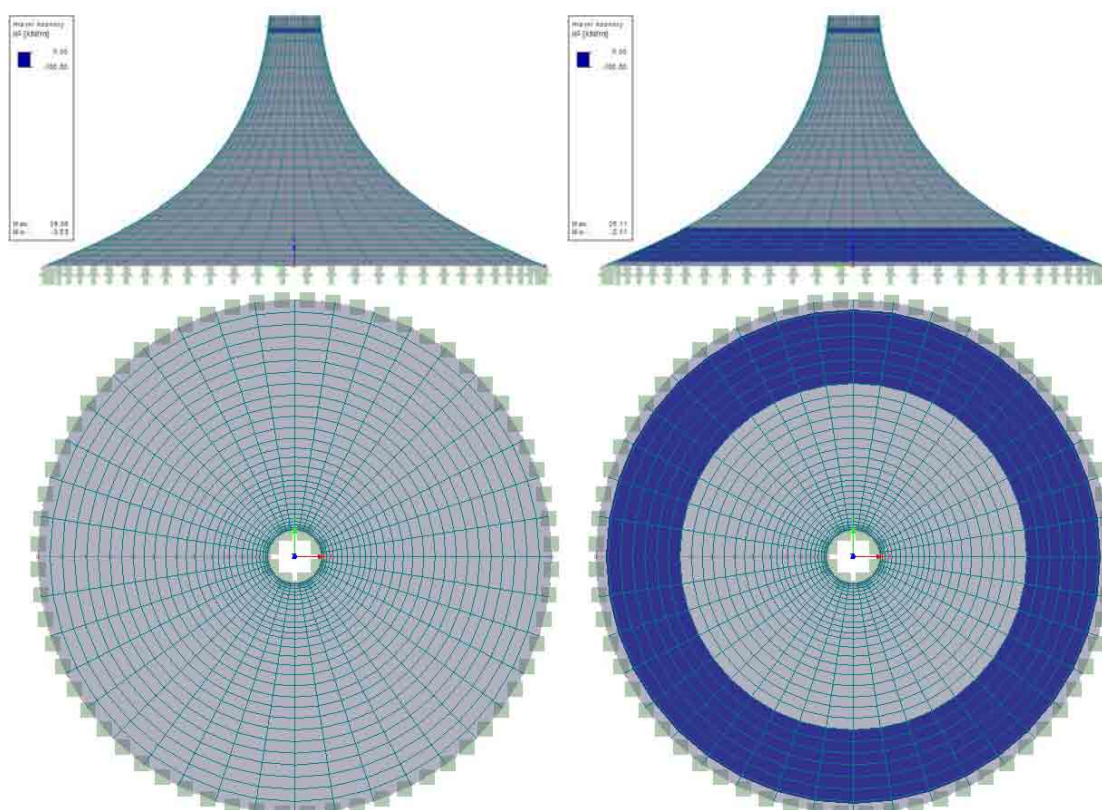
Na třetím tvaru opět vznikly deformace při stálém zatížení pouze o velikosti 10 mm. Při zatížení sněhem již v dolní části membrány vznikla maximální deformace o velikosti 12.5 mm. Hlavní napětí n_1 opět kleslo z 88.65 kN/m pro stálé zatížení na 59.47 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 vzrostly z -3.03 kN/m pro stálé zatížení na -2.11 kN/m pro kombinaci se sněhem. Pás příčných tlaků v dolní části membrány vznikl opět na větší ploše, zatímco příčné tlaky v horní části membrány se zmenšují. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 16 Deformace při stálém zatížení a kombinaci stálého se sněhem



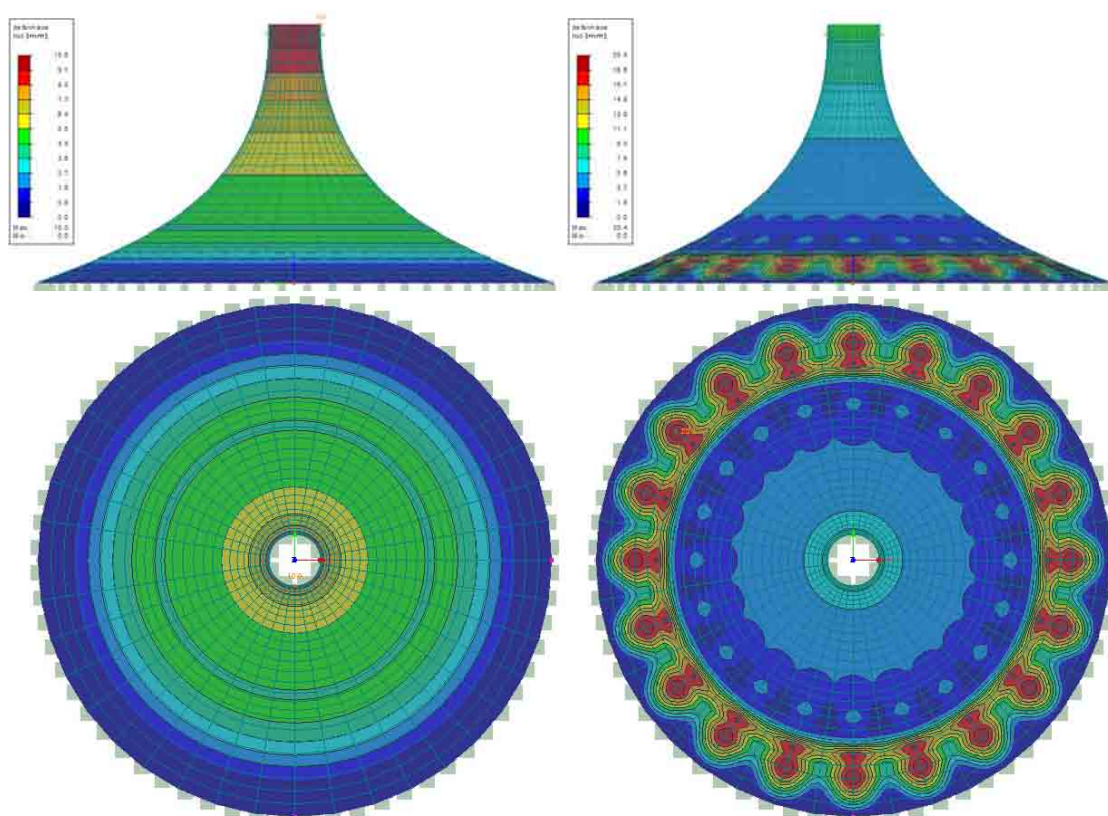
Obr. 17 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



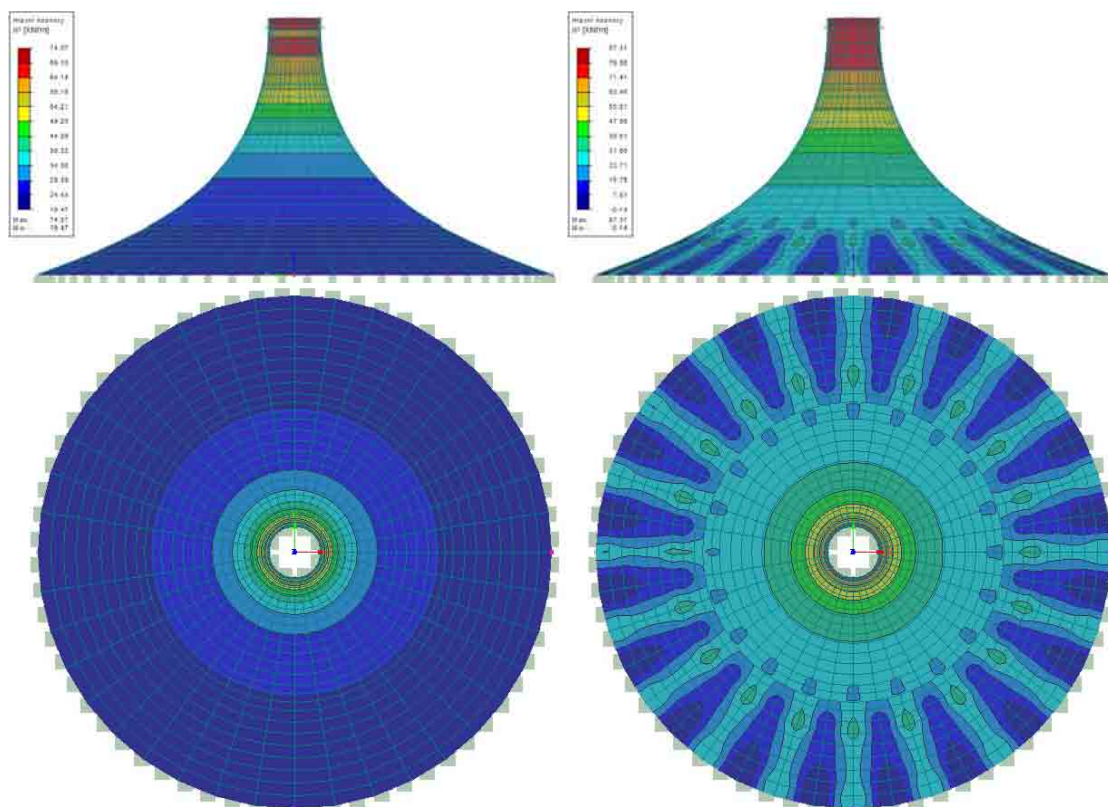
Obr. 18 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.1.4 Tvar 4

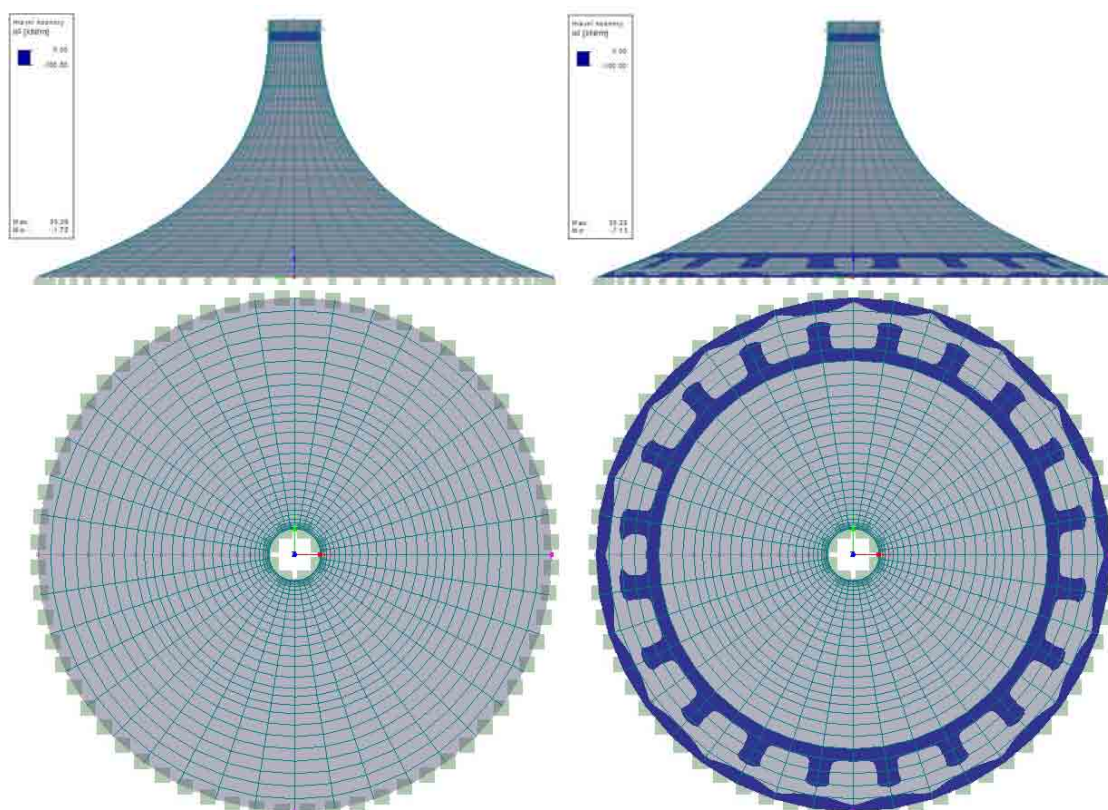
Čtvrtý tvar se při stálém zatížení opět deformuje o 10 mm. V kombinaci se sněhem pak vzrůstá deformace v dolní části membrány na 20.4 mm. U tohoto tvaru však již dochází k nerovnoměrné deformaci. Hlavní napětí n_1 vzrostlo ze 74.07 kN/m pro stálé zatížení na 87.31 kN/m v kombinaci se sněhem. Nejvyšší hodnoty hlavního napětí n_2 klesly z -1.72 kN/m pro stálé zatížení na -7.32 kN/m pro kombinaci se sněhem. V kombinaci se sněhem způsobil pás příčných tlaků v dolní části membrány pravidelné zvlnění membrány. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 19 Deformace při stálém zatížení a kombinaci stálého se sněhem



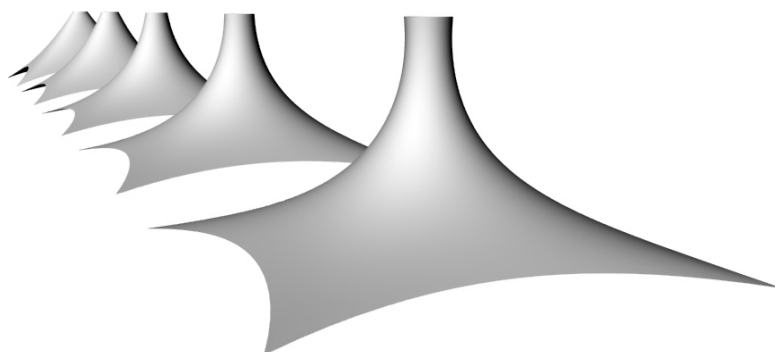
Obr. 20 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



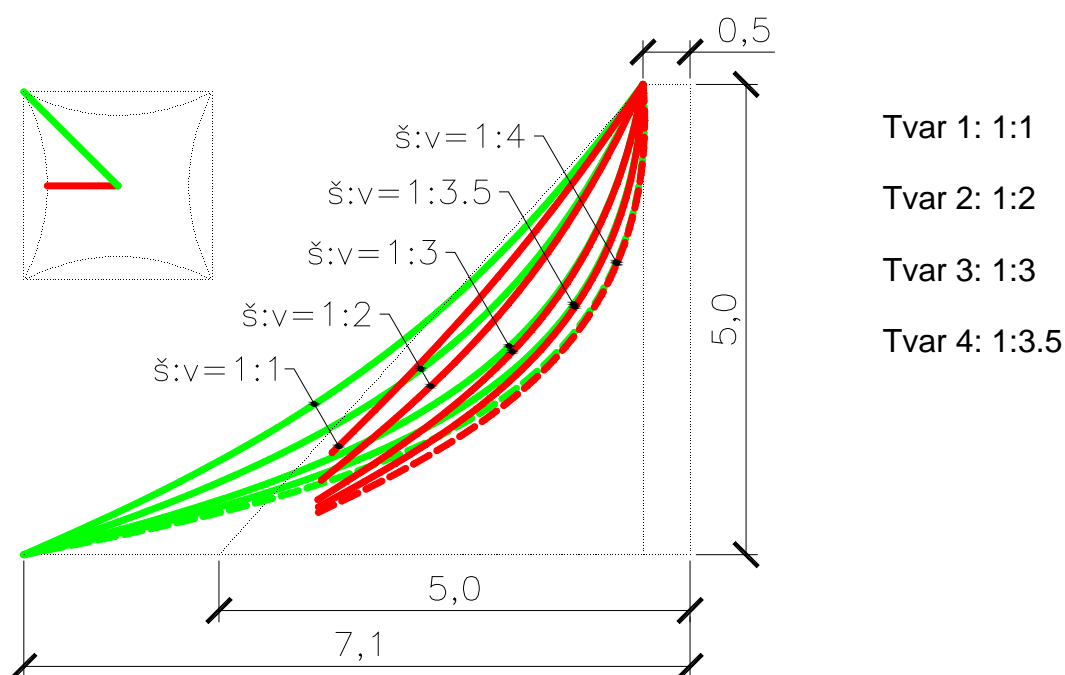
Obr. 21 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.2 Čtvercový půdorys

Vzhledem k tomu, že se nejedná o rotační ani cyklickou plochu, bylo nutné použít skript MeshRelaxation.rvb. Tento skript umožňuje dynamickou relaxaci sítí. Bohužel neumožňuje přiřazení různého napětí weft a warp. Proto jsem zvolil jiný přístup. Výchozí tvar sítě jsem namodeloval s různou výškou (celočíselným násobkem 5 m), avšak se stejným půdorysem. Po relaxaci jednotlivých sítí jsem provedl 1D zmenšení sítě ve směru osy z.



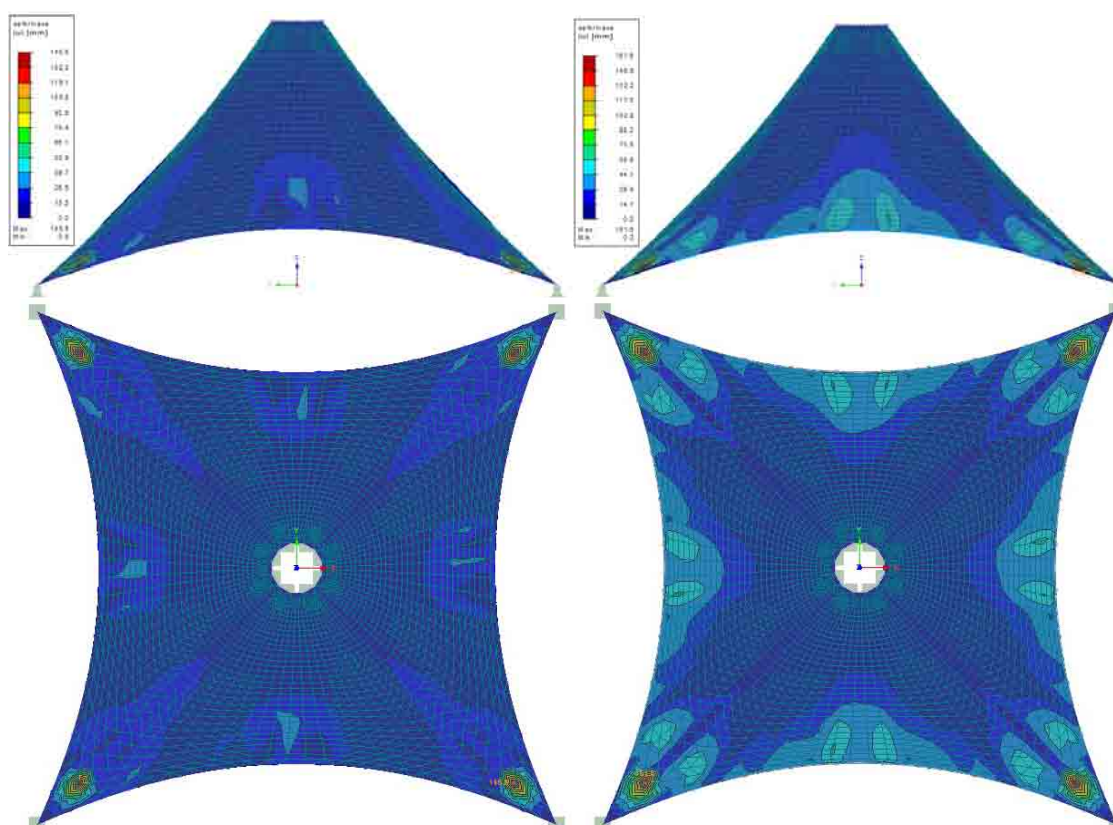
Obr. 22 Perspektiva analyzovaných tvarů



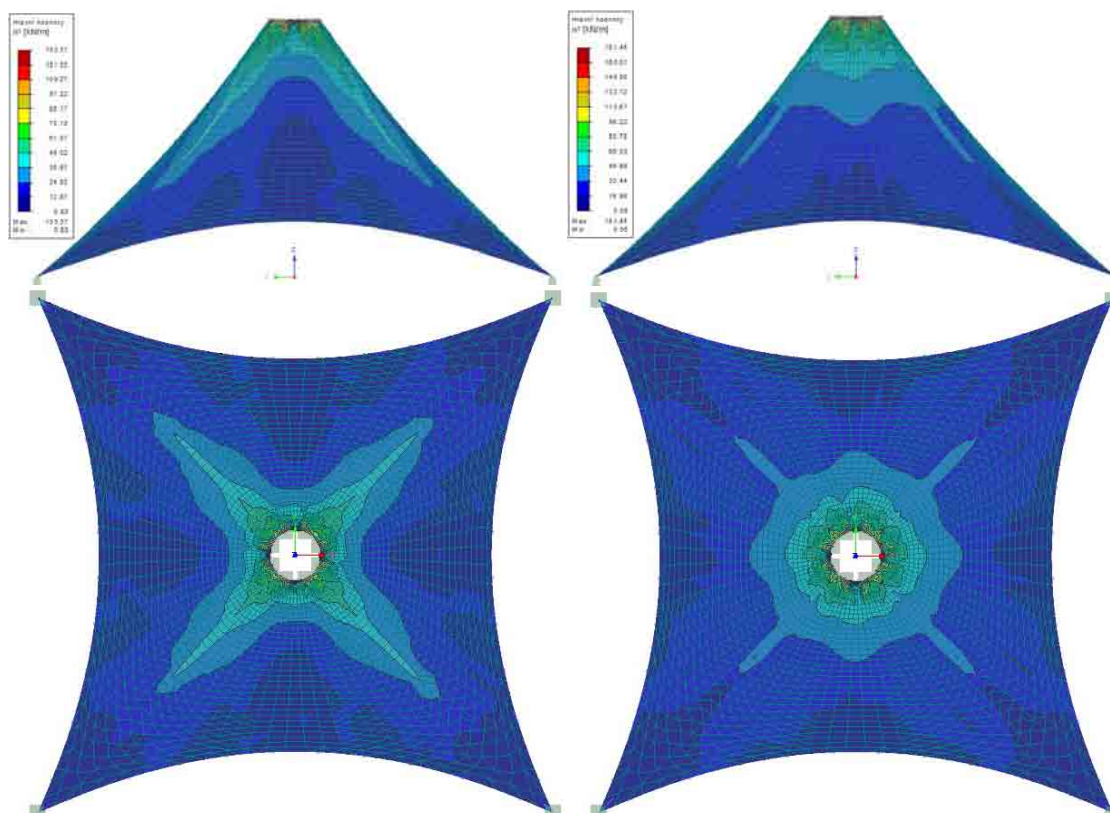
Obr. 23 Řezy tvary membrán čtvercového půdorysu

3.1.2.1 Tvar 1

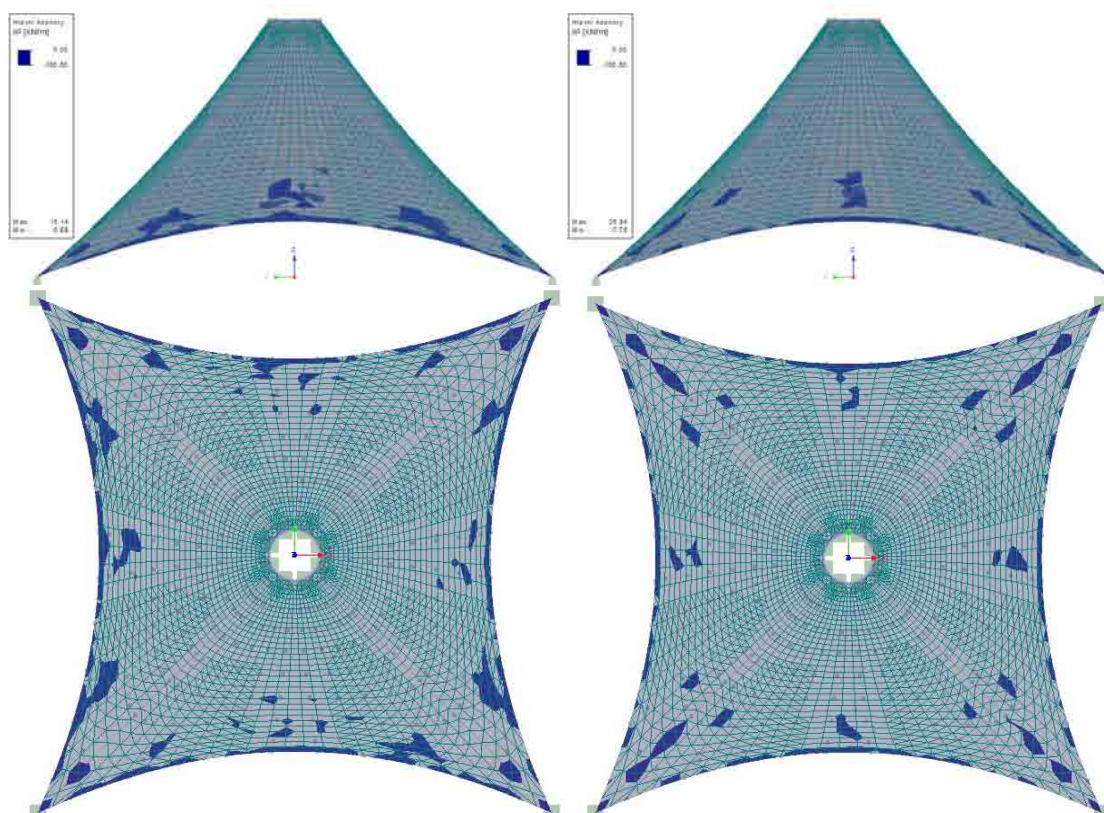
První tvar má při stálém zatížení deformaci o velikosti max.145.5 mm. V kombinaci se sněhem vzrostla maximální deformace na 161.6 mm. Hlavní napětí n_1 vzrostlo ze 133.37 kN/m pro stálé zatížení na 181.45 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -5.88 kN/m pro stálé zatížení na -7.78 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 24 Deformace při stálém zatížení a kombinaci stálého se sněhem



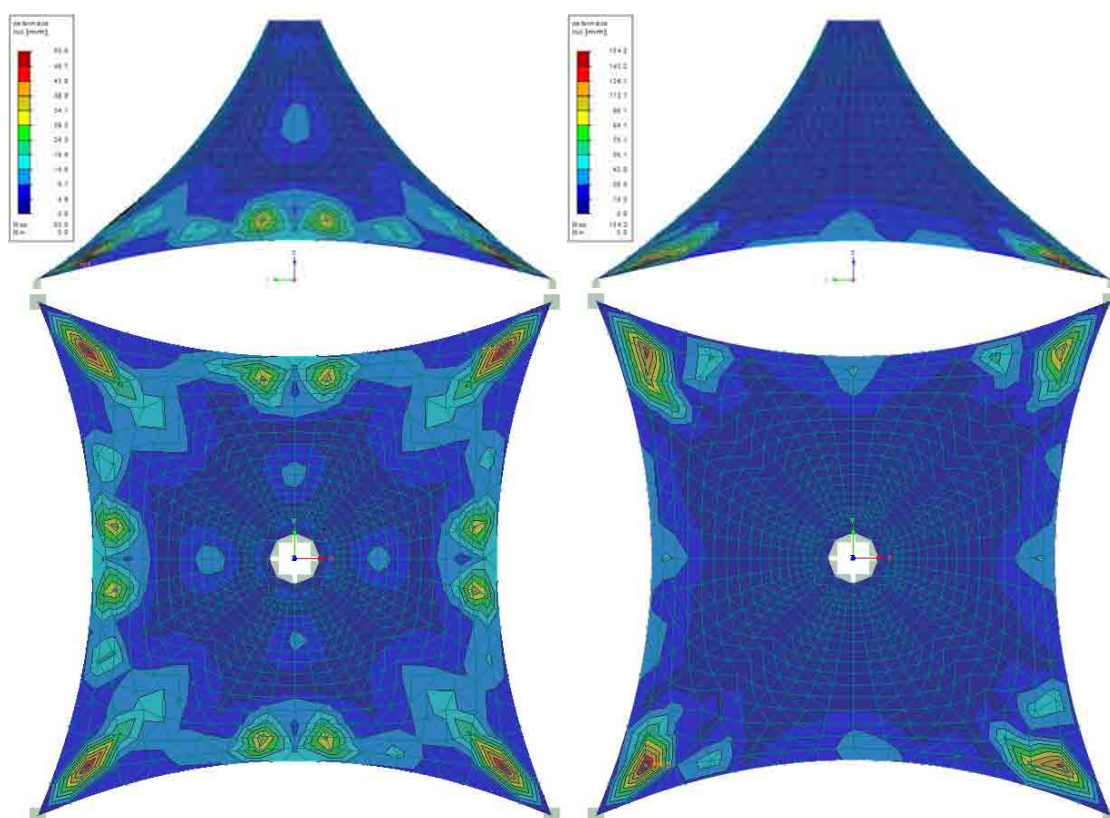
Obr. 25 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



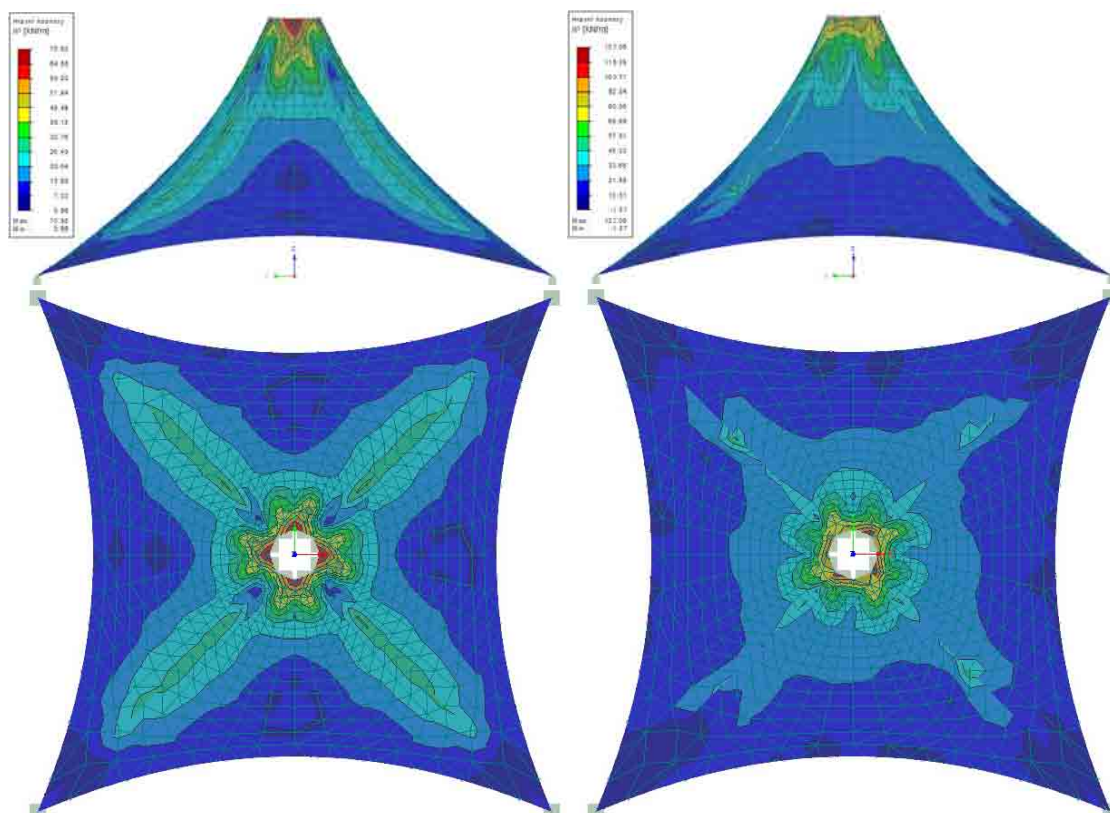
Obr. 26 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.2.2 Tvar 2

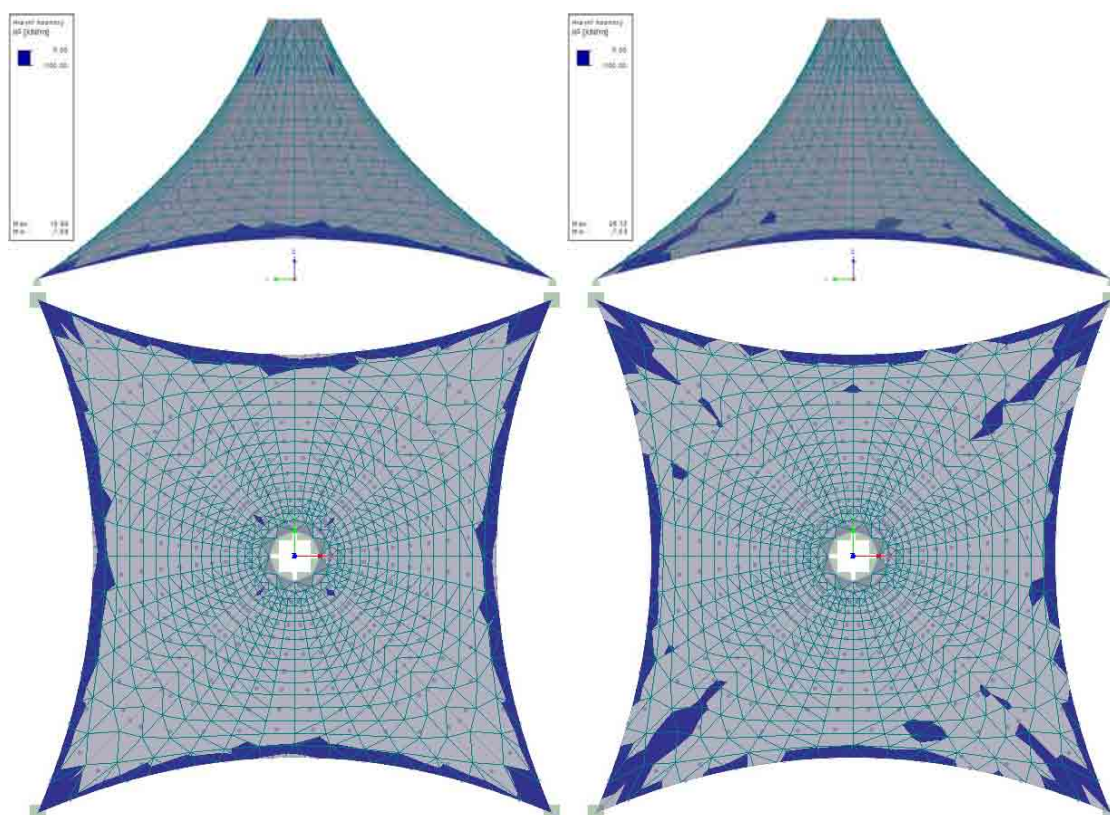
Maximální deformace na druhém tvaru činila při stálém zatížení 53.5 mm. V kombinaci se sněhem vzrostla maximální deformace na 154.2 mm. Hlavní napětí n_1 vzrostlo ze 70.92 kN/m pro stálé zatížení na 127.06 kN/m v kombinaci se sněhem. Bohužel se však při zatížení sněhem dostalo hlavní napětí n_1 až do záporných hodnot, konkrétně na -1.37 kN/m. Nejvyšší hodnoty hlavního napětí n_2 vzrostly z -7.08 kN/m pro stálé zatížení na -7.03 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 27 Deformace při stálém zatížení a kombinaci stálého se sněhem



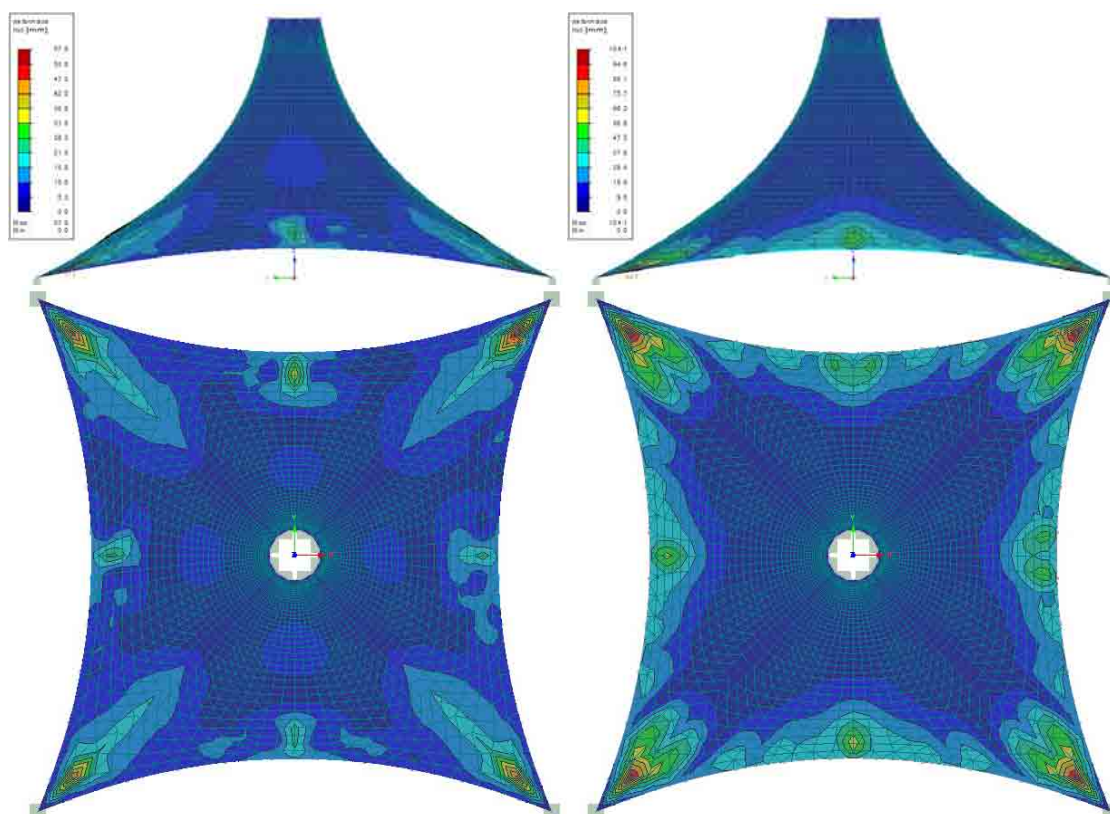
Obr. 28 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



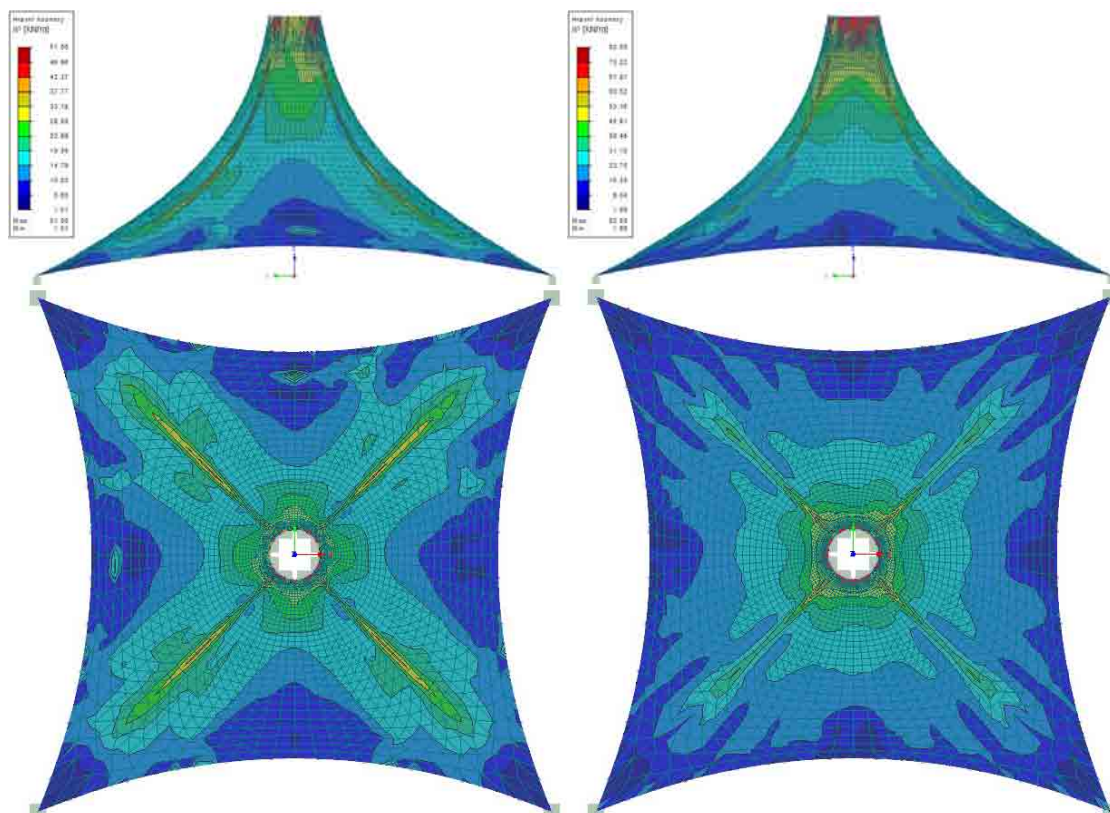
Obr. 29 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.2.3 Tvar 3

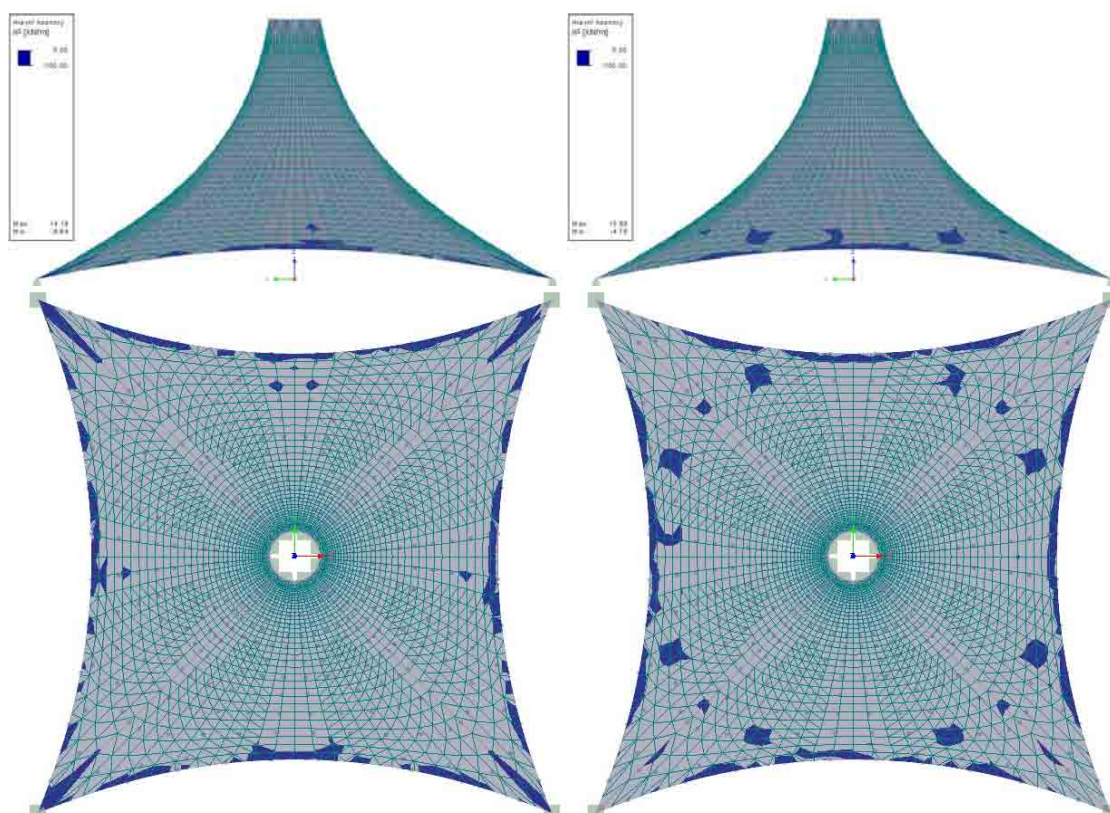
U třetího tvaru dosáhla maximální deformace při stálém zatížení hodnoty 57.8 mm. V kombinaci se sněhem vzrostla maximální deformace na 104.1 mm. Hlavní napětí n_1 vzrostlo ze 51.56 kN/m pro stálé zatížení na 82.58 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 se zvedly z -5.54 kN/m pro stálé zatížení na -4.78 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 30 Deformace při stálém zatížení a kombinaci stálého se sněhem



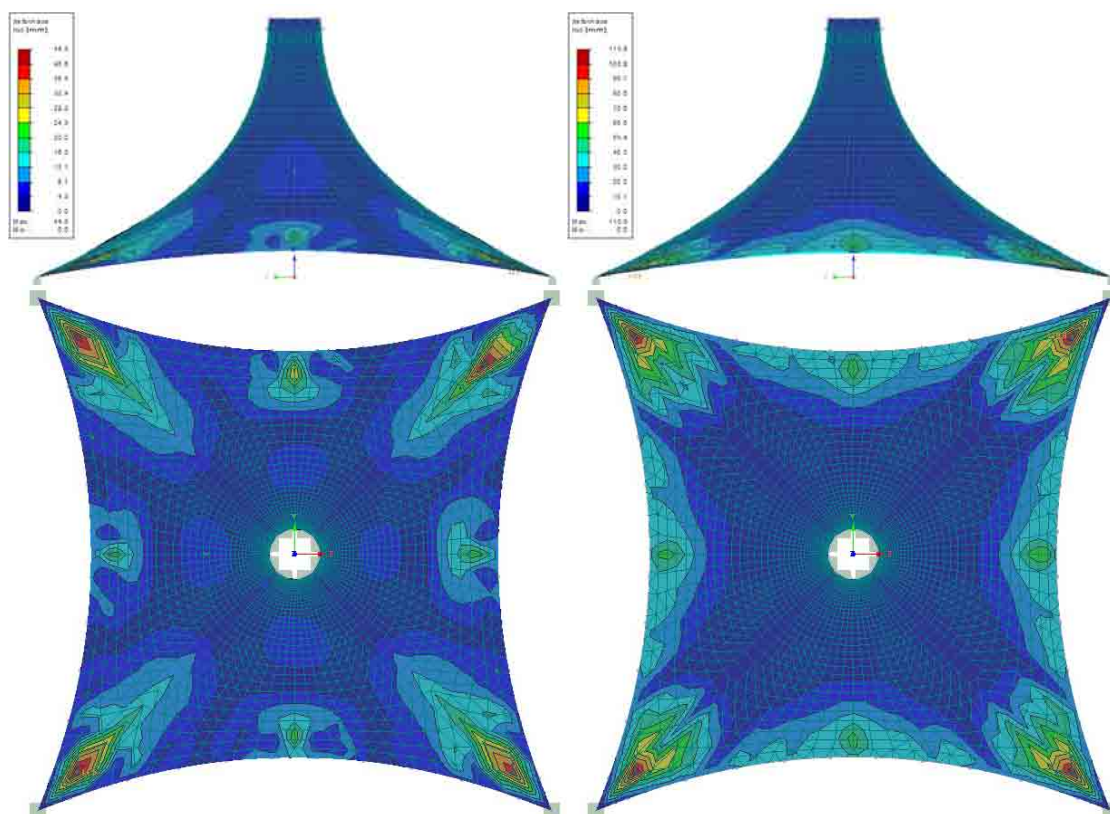
Obr. 31 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



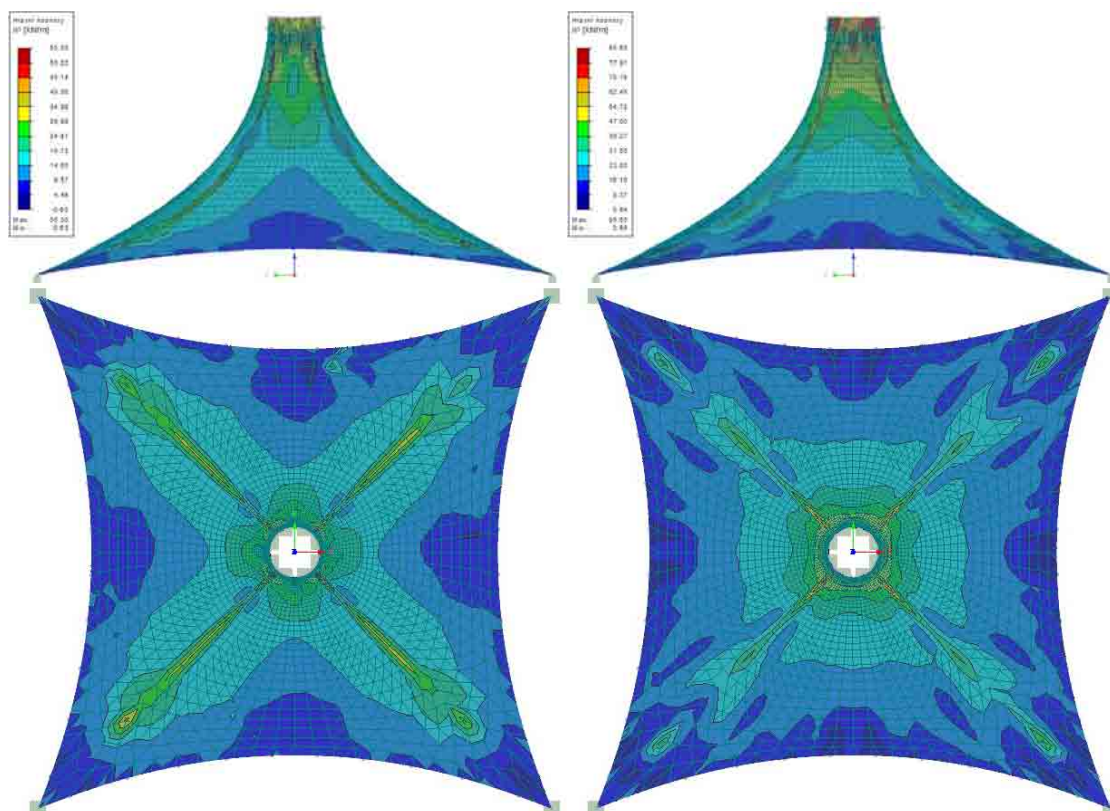
Obr. 32 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.2.4 Tvar 4

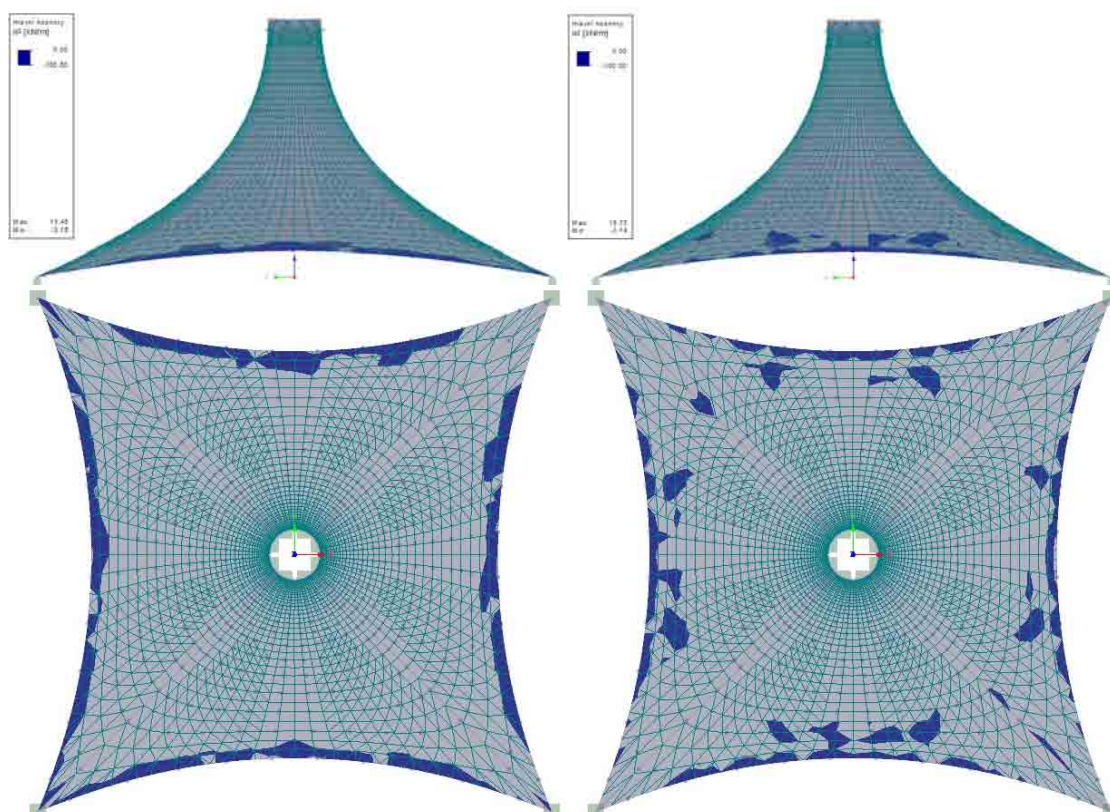
Poslední tvar má při stálém zatížení maximální deformaci o velikosti 44.5 mm. V kombinaci se sněhem vzrostla maximální deformace na 110.9 mm. Hlavní napětí n_1 vzrostlo ze 55.30 kN/m pro stálé zatížení na 85.63 kN/m v kombinaci se sněhem. Bohužel se v tomto případě opět hlavní napětí n_1 dostalo do záporných hodnot a to pro stálé zatížení, konkrétně až na hodnotu -0.60 kN/m. Nejnižší hodnoty hlavního napětí n_2 vzrostly z -3.16 kN/m pro stálé zatížení na hodnotu -3.14 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 33 Deformace při stálém zatížení a kombinaci stálého se sněhem



Obr. 34 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



Obr. 35 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.1.3 Shrnutí

Z analyzovaných tvarů rotačních ploch se jeví jako nejvýhodnější tvar s poměrem $weft:warp=1:3$ (Tvar 3). Při tomto poměru dochází k nejmenší oblasti tlakového napětí v horní části membrány. Přesto, že již dochází k deformaci v dolní části větší, než je deformace od posunutí podpor, nedochází zde ještě k boulení, které již nastává u tvaru 4.

V případě čtvercového půdorysu bych jako nejvhodnější zvolil opět tvar 3, který vykazoval nejmenší deformaci při kombinaci stálého a nahodilého zatížení. Zároveň hlavní napětí n_1 nenabývalo záporných hodnot. Hlavní napětí n_2 nabývalo záporných hodnot převážně v oblastech kolem lanových prutů. Tyto pruty jsou soudržně spojeny s přilehlými plochami. Proto předpětí vnesené do prutů způsobuje tlakové namáhání přilehlé membrány rovnoběžně s osou prutu.

Dále je možné si povšimnout koncentrací napětí kolem spojů jednotlivých NURBS ploch, ze kterých je konstrukce vymodelována. Je to způsobeno rozdílem mezi geometrií získanou pomocí Rhino skriptů a výslednou, složenou ze čtyř NURBS ploch.

3.2 Tvary vytvořené v programu Formfinder

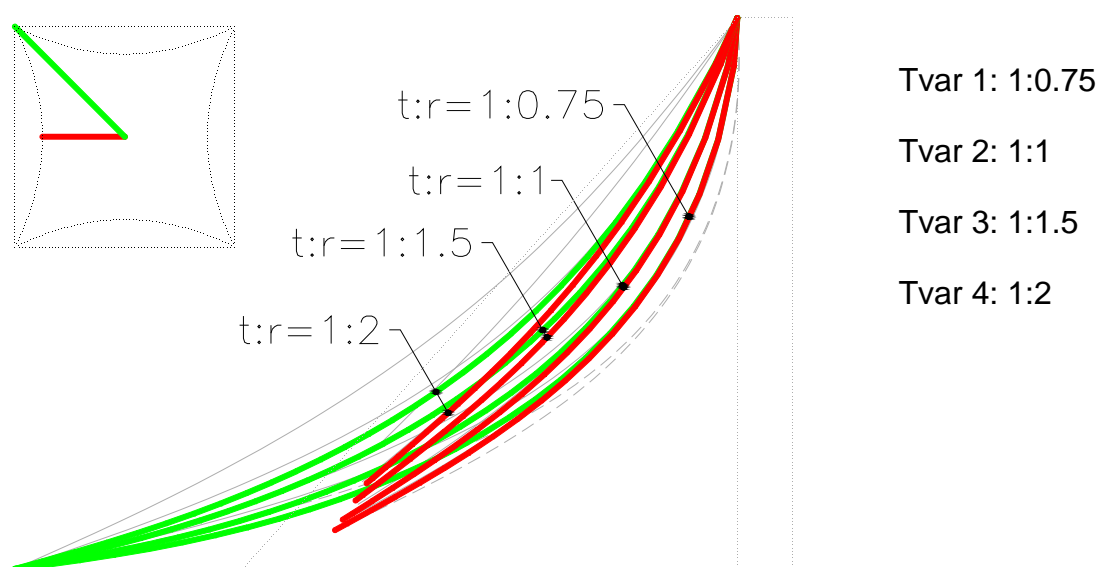
Program FormFinder klade důraz na jednoduchost návrhu tvaru a proto je jeho prostředí velice uživatelsky přívětivé. Podstatou návrhu je nakreslení půdorysného tvaru včetně možnosti ovlivnit velikost prohnutí obvodových předpínacích lan.

Pokud modelujeme membránu uchycenou pouze v rozích, zvolíme pravidelnou síť. V případě, že je požadavek na podporu uvnitř půdorysu membrány, je potřeba zvolit radiální síť. Ihned je vygenerována vnitřní kruhová podpora s předdefinovaným poloměrem. Výška podpory je nulová a proto je třeba zadat požadovanou výšku, případně poloměr podpory. V případě, že je potřeba více podpor, je možné vymodelovat více membrán s radiální sítí a ty pak spojit dohromady. Bohužel tato funkce je dostupná až v plné verzi.

Tento program umožňuje nastavení napětí pro weft a warp přímo. Způsob výpočtu lze ovlivnit i volbou metody výpočtu sil/napětí v radiálním směru. Jsou možné tři volby:

- konstantní napětí (akademický přístup)
- proměnná síla s počáteční hodnotou
- konstantní síla (konvenční přístup).

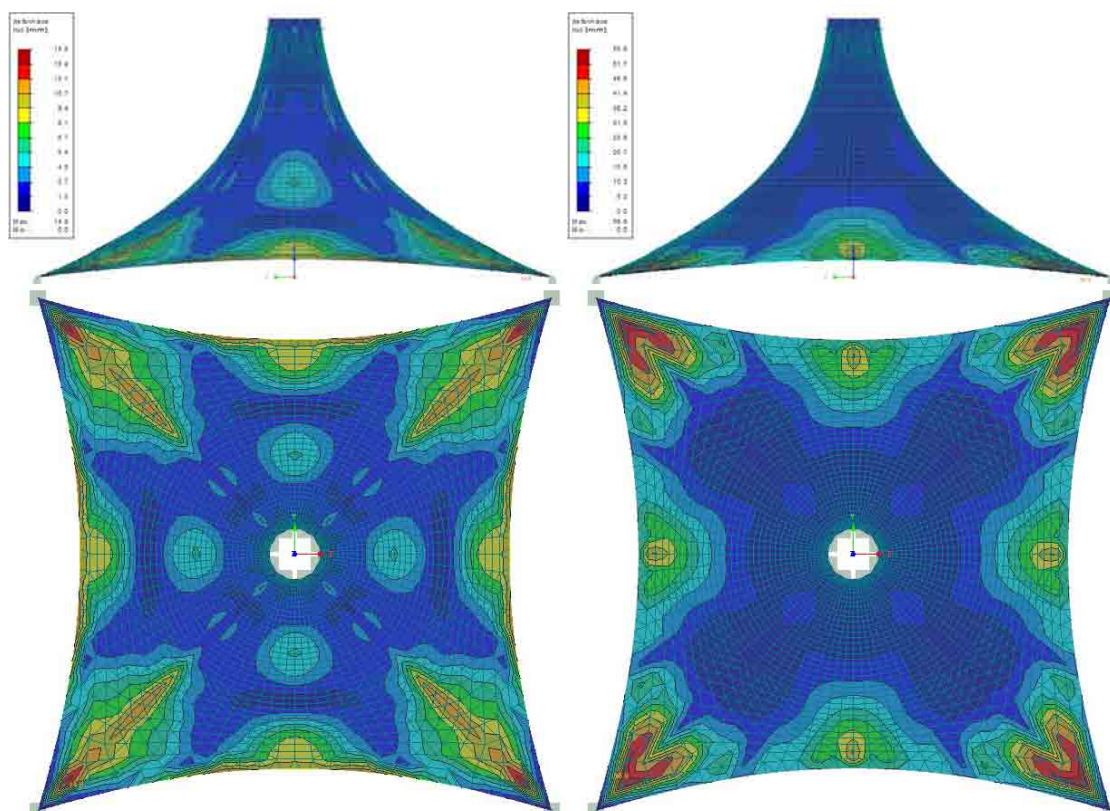
Tím lze ovlivnit výsledný tvar tak, aby splňoval požadavky architekta.



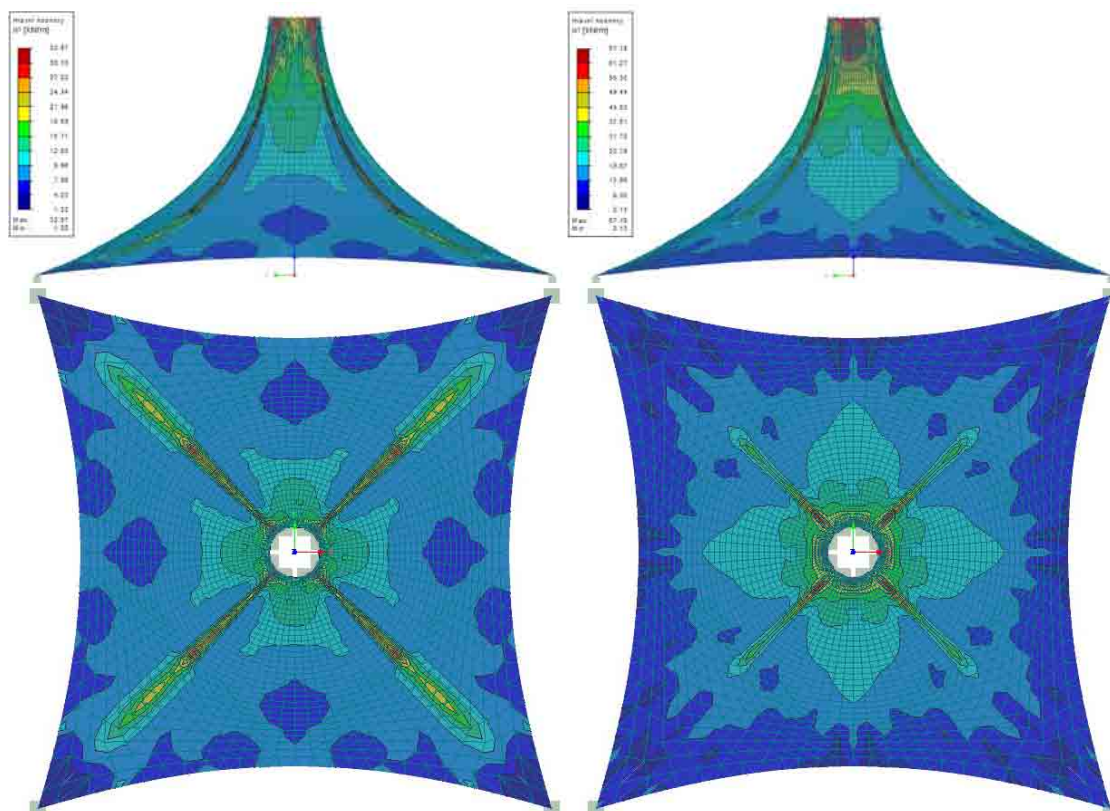
Obr. 36 Řezy s různým poměrem t:r

3.2.1.1 Tvar 1

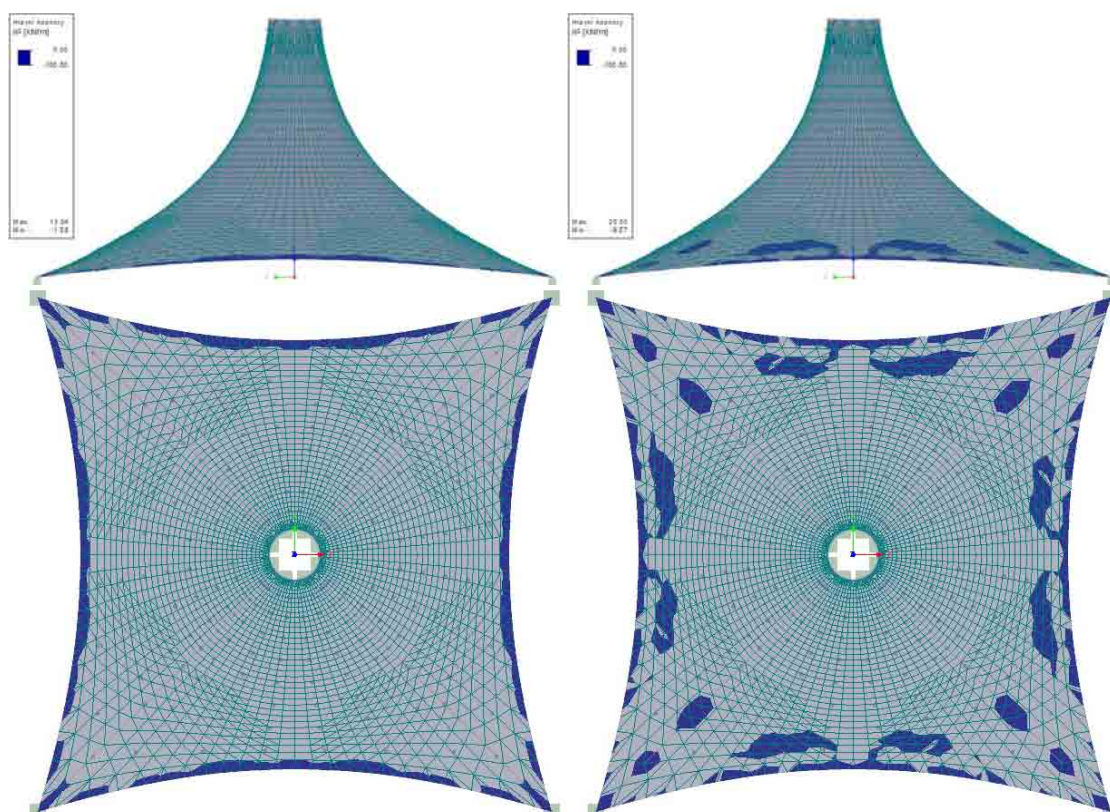
První tvar má při stálém zatížení maximální deformaci 14.8 mm. V kombinaci se sněhem vzrostla maximální deformace na 56.9 mm. Hlavní napětí n_1 vzrostlo ze 32.97 kN/m pro stálé zatížení na 67.18 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -1.28 kN/m pro stálé zatížení na -9.27 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 37 Deformace při stálém zatížení a kombinaci stálého se sněhem



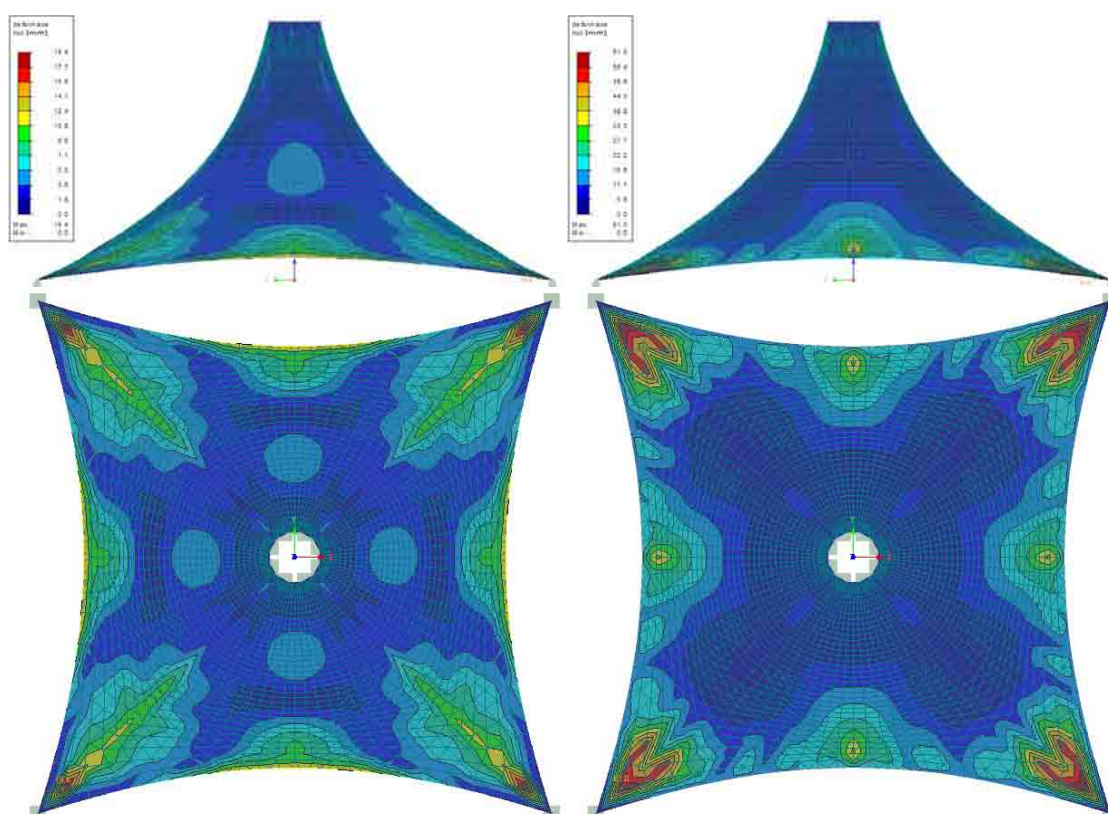
Obr. 38 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



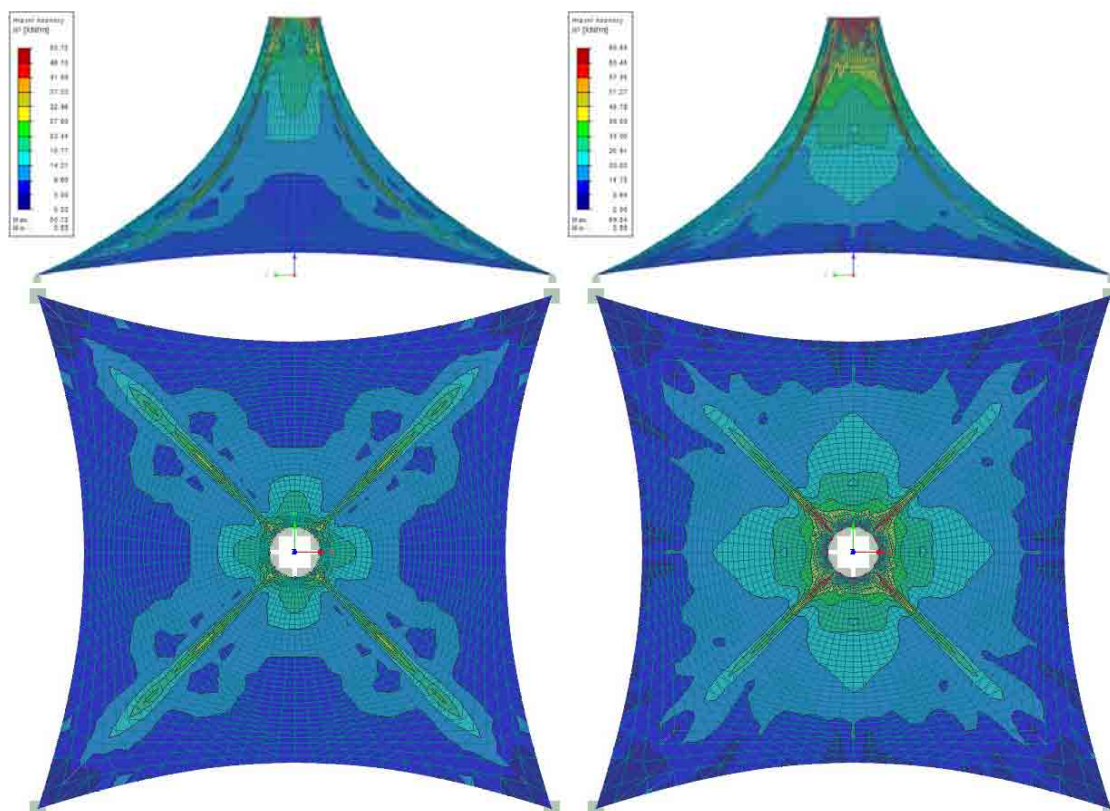
Obr. 39 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.2.1.2 Tvar 2

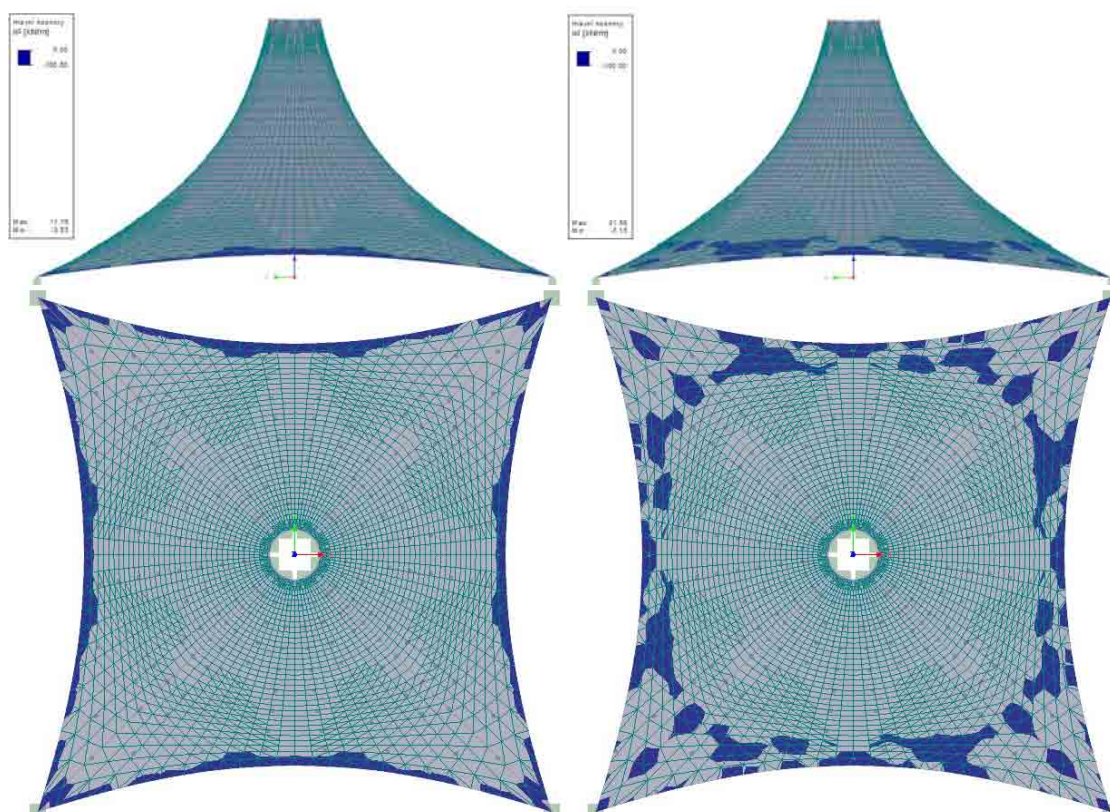
Druhý tvar má při stálém zatížení deformaci o velikosti max.19.4 mm. V kombinaci se sněhem vzrostla maximální deformace na 61.0 mm. Hlavní napětí n_1 vzrostlo ze 50.72 kN/m pro stálé zatížení na 69.54 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -3.83 kN/m pro stálé zatížení na -5.16 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 40 Deformace při stálém zatížení a kombinaci stálého se sněhem



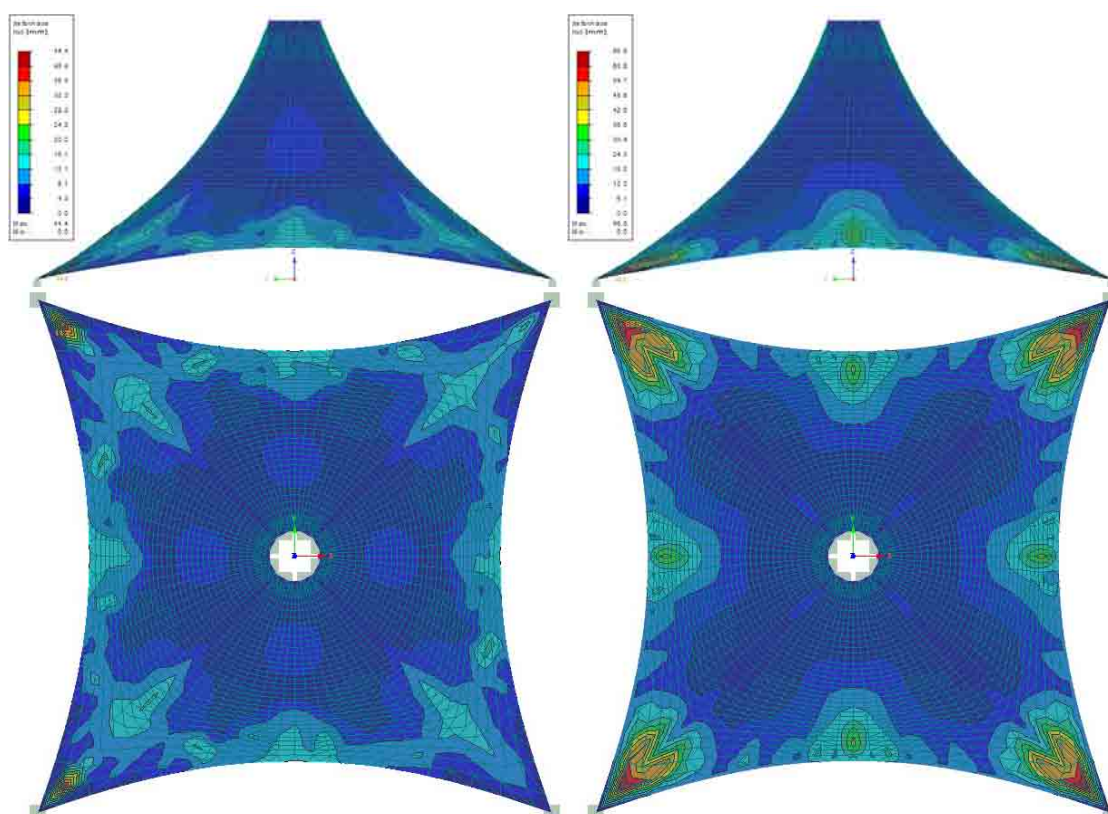
Obr. 41 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



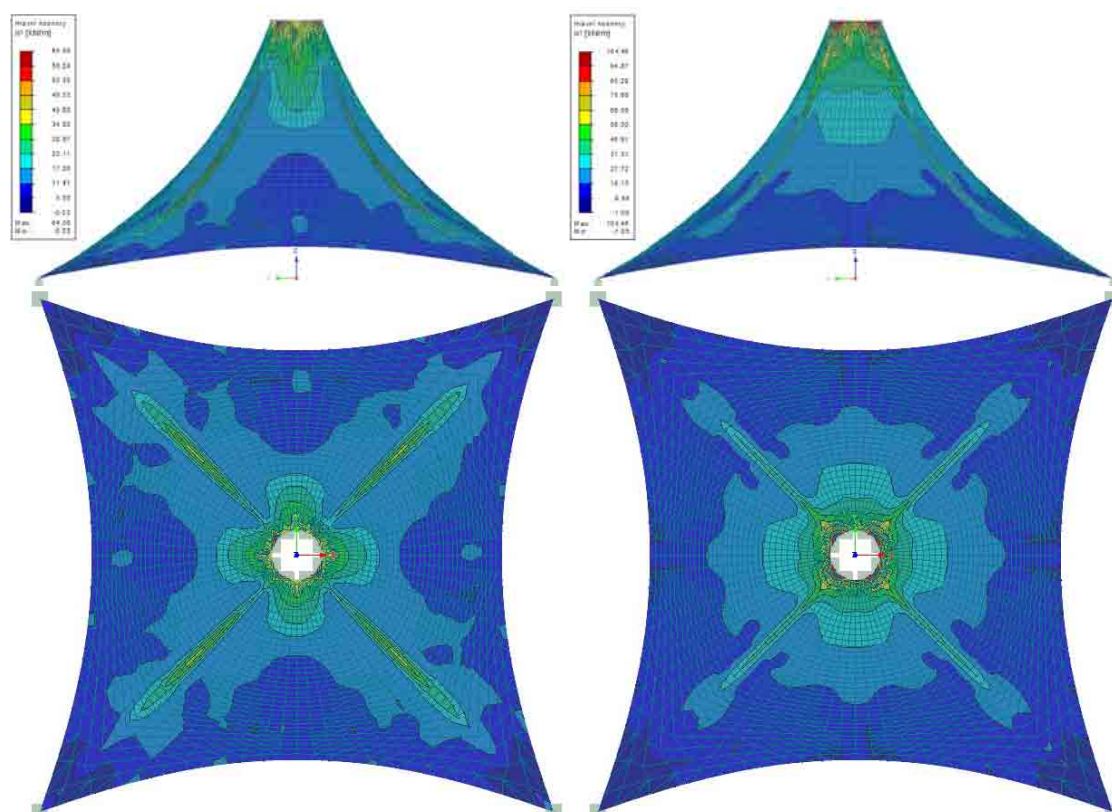
Obr. 42 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.2.1.3 Tvar 3

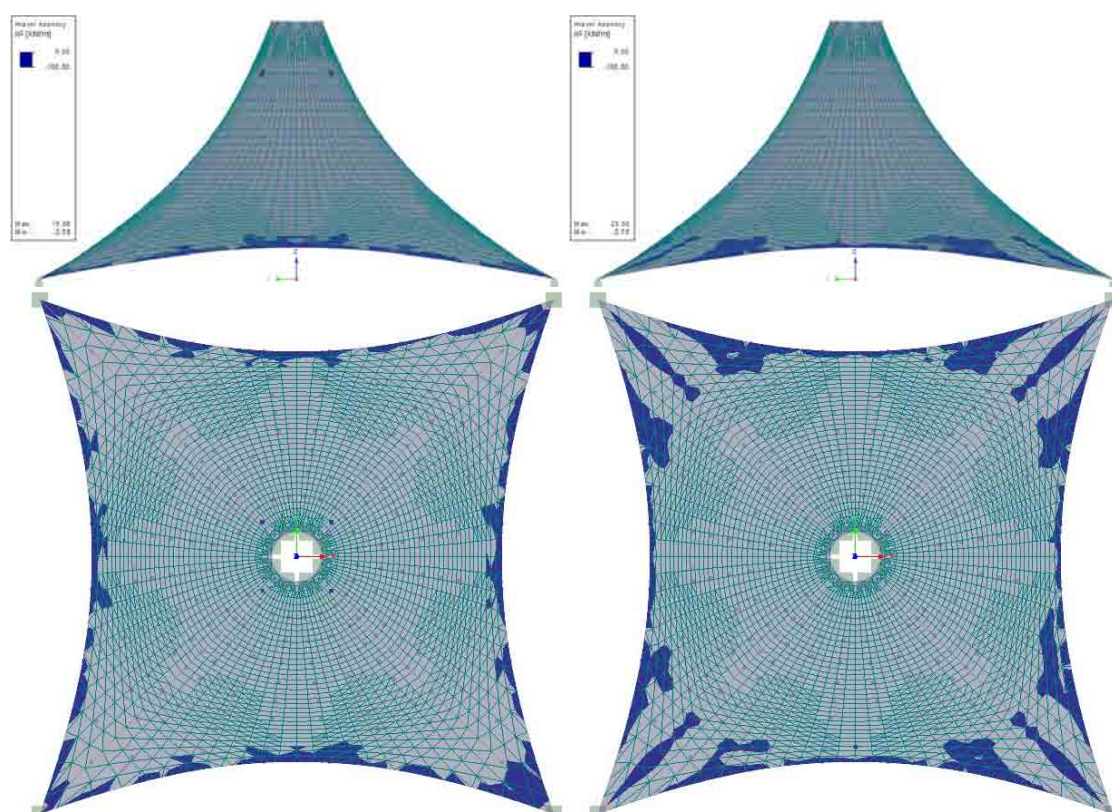
Třetí tvar při stálém zatížení nabývá maximální deformace 44.4 mm. V kombinaci se sněhem vzrostla maximální deformace na 66.8 mm. Hlavní napětí n_1 vzrostlo ze 64.09 kN/m pro stálé zatížení na 104.46 kN/m v kombinaci se sněhem. Bohužel v tomto případě opět hlavní napětí n_1 nabývalo záporných hodnot a to -0.30 kN/m pro stálé zatížení a -1.06 kN/m pro kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 vzrostly z -2.39 kN/m pro stálé zatížení na -2.15 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 43 Deformace při stálém zatížení a kombinaci stálého se sněhem



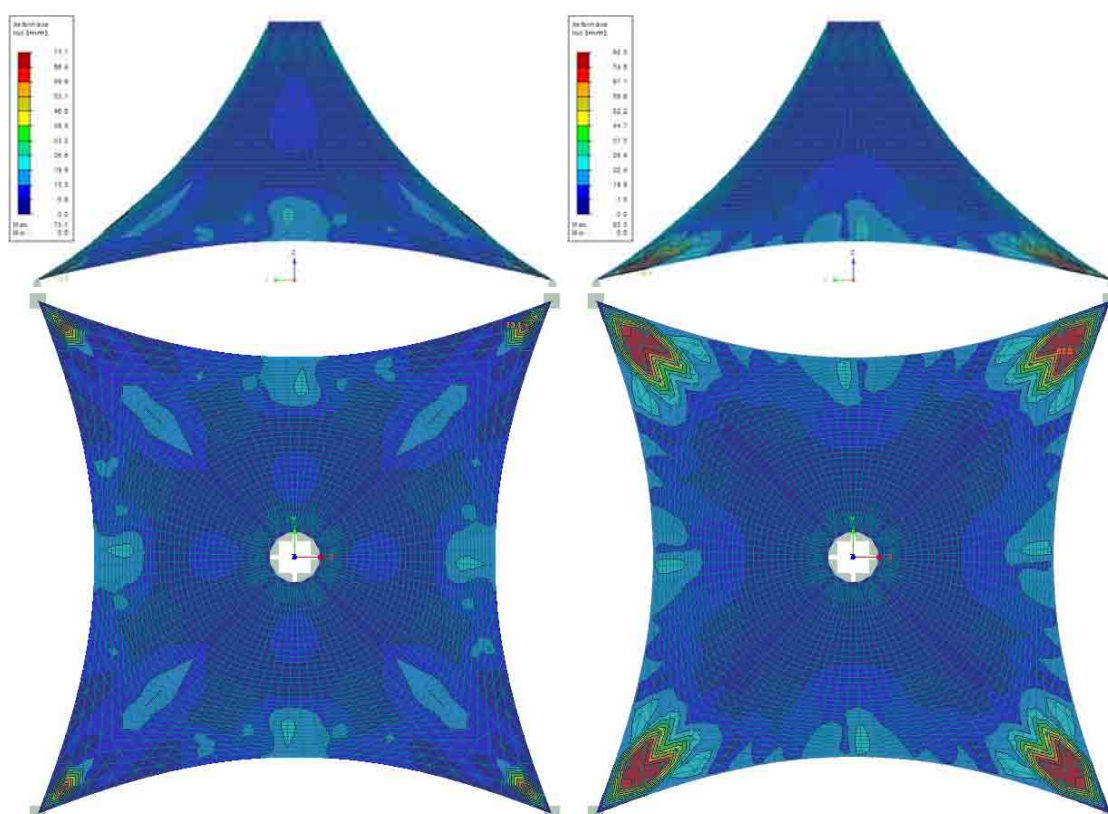
Obr. 44 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



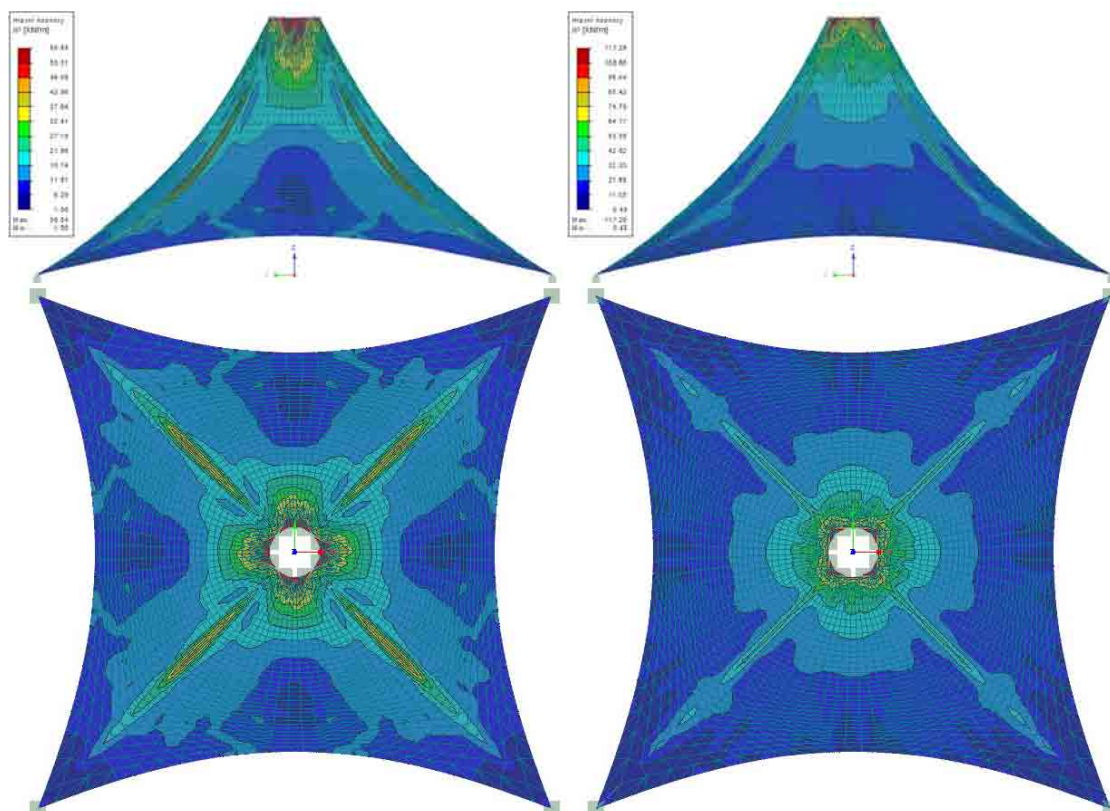
Obr. 45 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.2.1.4 Tvar 4

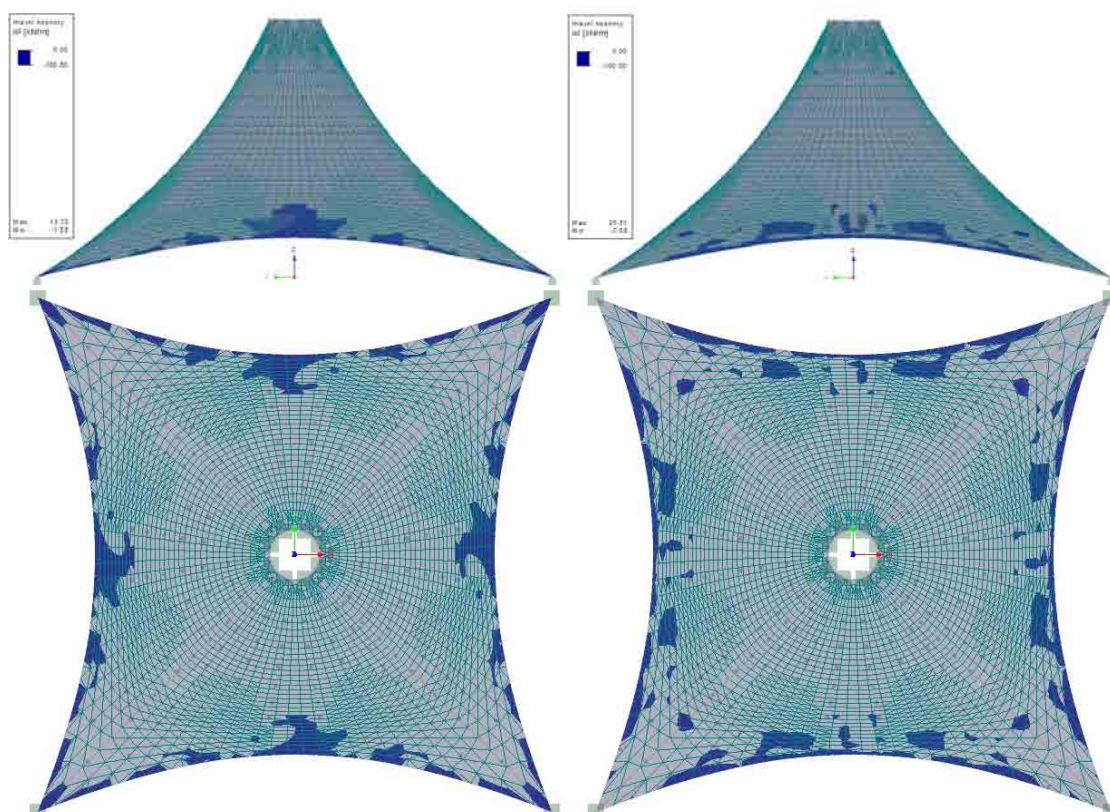
Pro čtvrtý tvar při stálém zatížení dosahuje maximální deformace 73.1 mm. V kombinaci se sněhem vzrostla maximální deformace na 82.0 mm. Hlavní napětí n_1 vzrostlo ze 58.54 kN/m pro stálé zatížení na 117.29 kN/m v kombinaci se sněhem. Nejvyšší hodnoty hlavního napětí n_2 vzrostly z -1.25 kN/m pro stálé zatížení na -2.38 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 46 Deformace při stálém zatížení a kombinaci stálého se sněhem



Obr. 47 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



Obr. 48 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.2.2 Shrnutí

Z vybraných tvarů získaných pomocí programu FormFinder bych jako nejvhodnější zvolil tvar 1. A to z důvodu, že vykazuje nejmenší deformace. Zároveň na konstrukci vznikají nejmenší hlavní napětí n_1 . Přítomnost záporného hlavního napětí n_2 je opět způsobena z velké části předpínacími lanovými pruty.

Z provedených analýz vyplývá, že veškeré tvary získané pomocí programu FormFinder jsou vhodné a dobře použitelné pro membránové konstrukce.

3.3 Hledání tvaru programem RFEM

Program RFEM od firmy Dlubal je efektivním softwarem pro analýzu konstrukcí a to především se zohledněním nelineárních účinků. Umožňuje většinu možných způsobů zatížení a proto je pro inženýrskou praxi nepostradatelným pomocníkem. Ačkoliv není přímo určen k hledání vhodného tvaru membránových konstrukcí, využiji jeho možností nelineárního výpočtu k hledání vhodných tvarů membránových konstrukcí.

Metoda Updated Reference Strategy použitá v programu RhinoMembrane totiž také vychází přímo z teorie nelineární mechaniky. Program RFEM využívá teorie Updated Lagrangián, která je formulována na běžné konfiguraci (tj. v prostorových souřadnicích) a při výpočtu Newton-Raphsonovu iterační metodu.

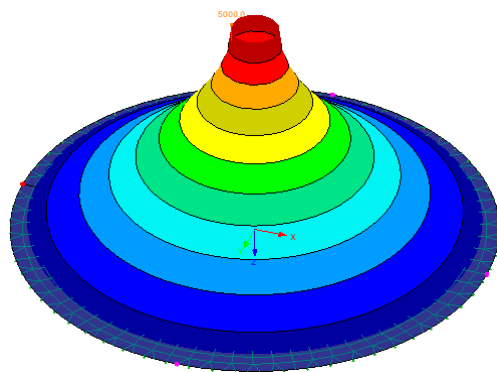
Abych dosáhl požadovaného výsledku, provedu několik analýz různými způsoby, které program RFEM umožňuje.

3.3.1 Vynucený posun podpor

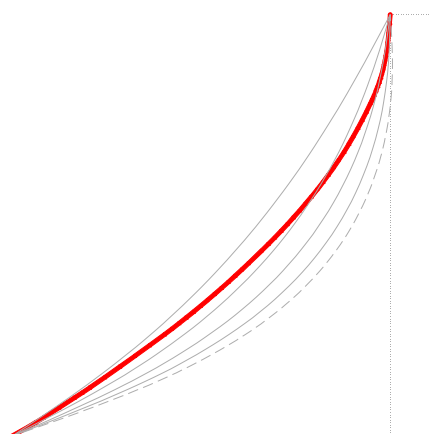
První možností, jak vytvořit plochu membrány, je vynucená deformace podpor. Nejdříve se vymodeluje počáteční tvar membrány jako 2D plocha. Tato plocha se následně zatíží vynuceným přemístění podpor do jejich výsledných prostorových souřadnic.

3.3.1.1 Rotační plocha

Jako počáteční tvar této konstrukce bylo použito mezikruží s vnějším poloměrem 5 m a vnitřním poloměrem 0.5 m v rovině XY. Vnější a vnitřní obvod byl opatřen liniovou podporou. Vnitřní liniová podpora byla následně zatížena vynuceným posunutím podporových linií o 5 m kolmo k rovině XY.



Obr. 49 Tvar získaný vynuceným posunutím vnitřní podpory



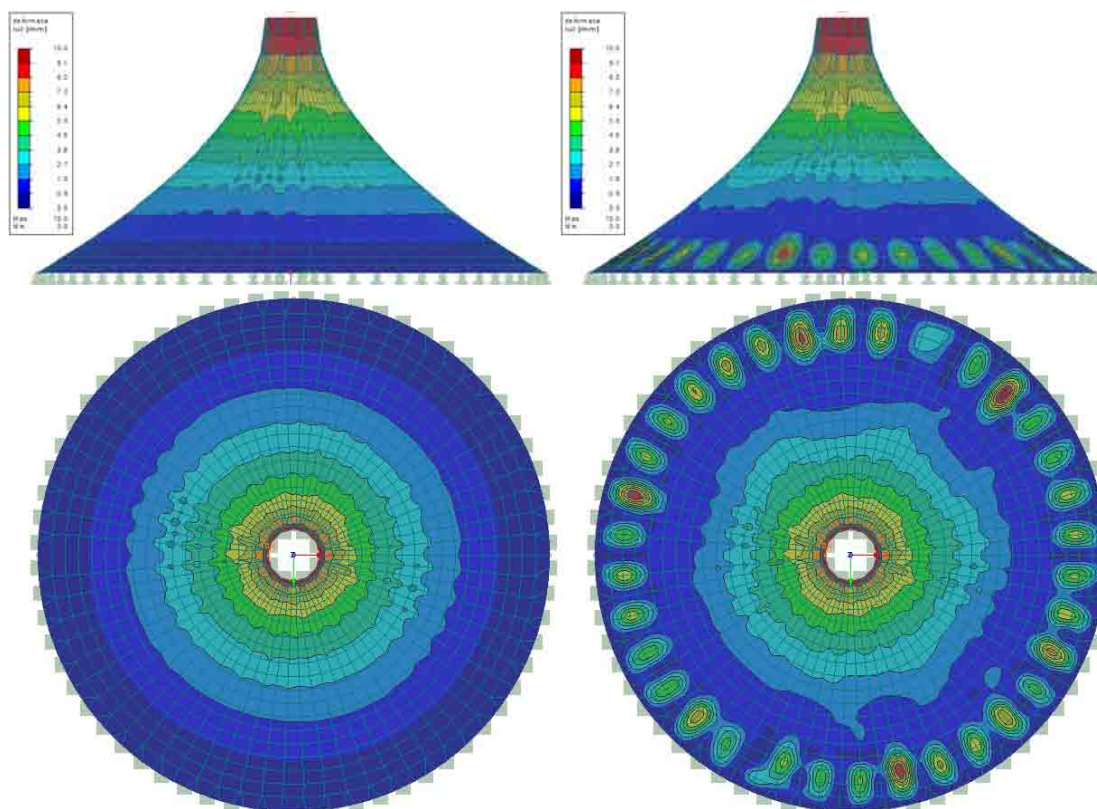
Obr. 50 Řez plochou a porovnání s řezy z kapitoly 4.1.1

Pro názornost budu všechny tvary porovnávat s tvary získanými pomocí relaxace rotační NURBS plochy.

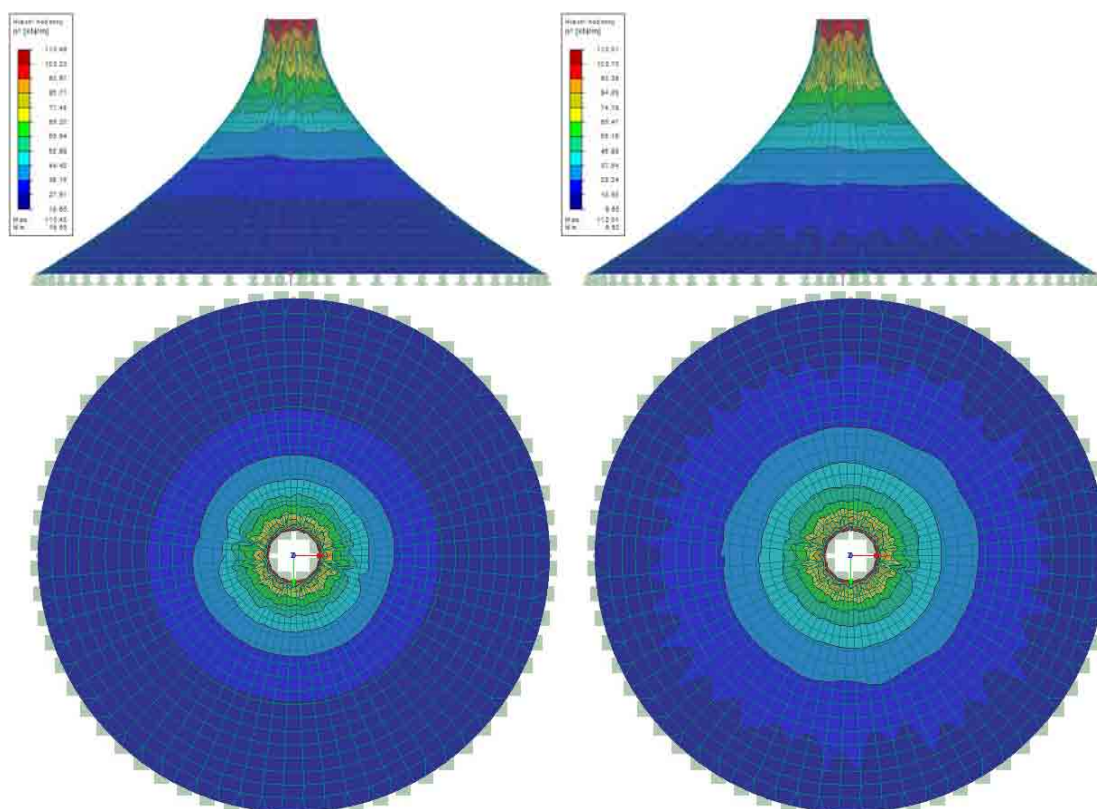
Při následné analýze se daný tvar choval podobně jako tvary získané pomocí Rhino skriptů. Při zatížení stálým zatížením nevznikla větší deformace než ta, která byla dána vynuceným posunutím vnitřní podpory. V kombinaci se sněhem se deformace též pohybovaly v rozmezí do 10 mm, avšak v dolní části membrány začalo vznikat opět boulení od příčného tlaku.

Co ale na rozdíl od ostatních tvarů nevznikalo, byly tlaky v horní části membrány. Při zatížení stálým zatížením dokonce nevznikaly žádné tlaky a celá membrána byla tudíž pouze tažena.

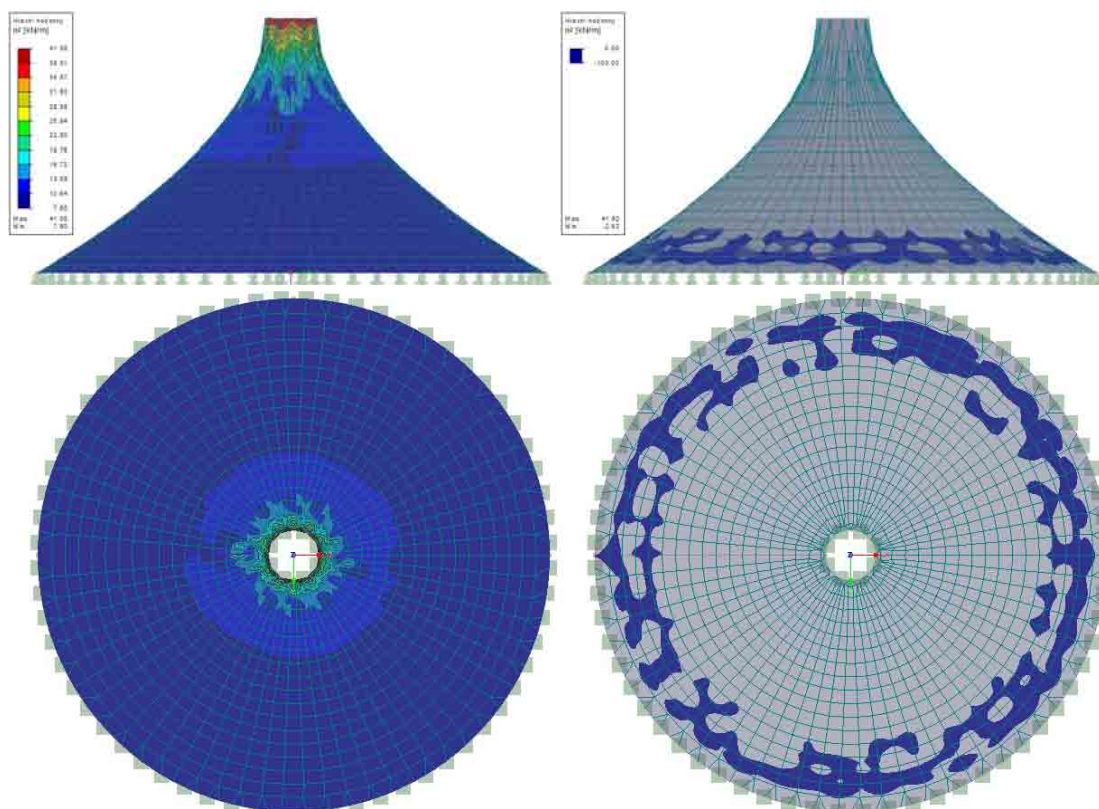
Hlavní napětí n_1 vzrostlo ze 110.48 kN/m při stálém zatížení na 112.01 kN/m při kombinaci se sněhem. Hlavní napětí n_2 kleslo ze 7.60 kN/m při stálém zatížení na -2.92 kN/m při kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 51 Deformace při stálém zatížení a kombinaci stálého se sněhem



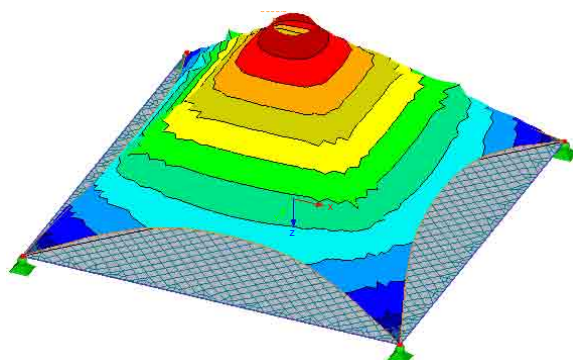
Obr. 52 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



Obr. 53 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.1.2 Čtvercový půdorys

Jako počáteční tvar této konstrukce byla zvolena čtvercová plocha v rovině XY o rozměrech 10 x 10 m s kruhovým otvorem ve středu o poloměru 0.5 m. Po vnějším obvodě byl umístěn prut s průřezem a charakteristikami lana PG 20 od firmy Pfeifer. Rohy byly osazeny bodovými podporami a vnitřní obvod liniovou podporou. Vnitřní liniovou podporu jsem následně zatížil vynuceným posunutím podporových linií o 5 m kolmo k rovině XY.



Obr. 54 Deformace čtvercového tvaru při vynuceném posunu podpor

Jak je zřejmé z obr. 54, tak v diagonálách dochází k vrásnění povrchu. Proto není tento způsob vhodný a nebudu jej proto ani analyzovat.

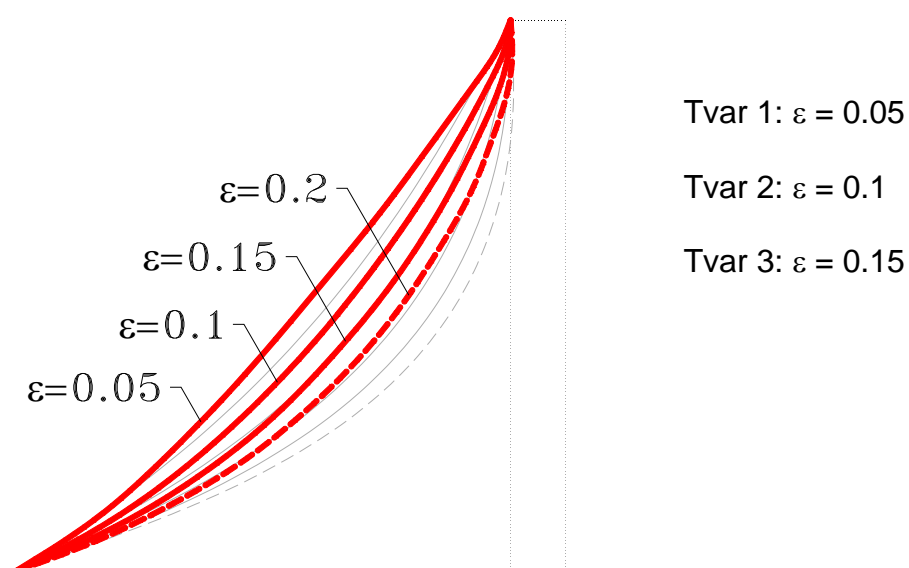
3.3.2 Předpětí v ploše pomocí smrštění

Druhou možností nalezení vhodného tvaru je zatížení počátečního tvaru poměrným přetvořením.

3.3.2.1 Rotační plocha

3.3.2.1.1 Konstantní poměrné přetvoření na celé ploše

Pro porovnání aplikuji na plochu zatížení poměrným přetvořením a to o 5, 10, 15 a 20%. Toto přetvoření působí v obou směrech stejně.

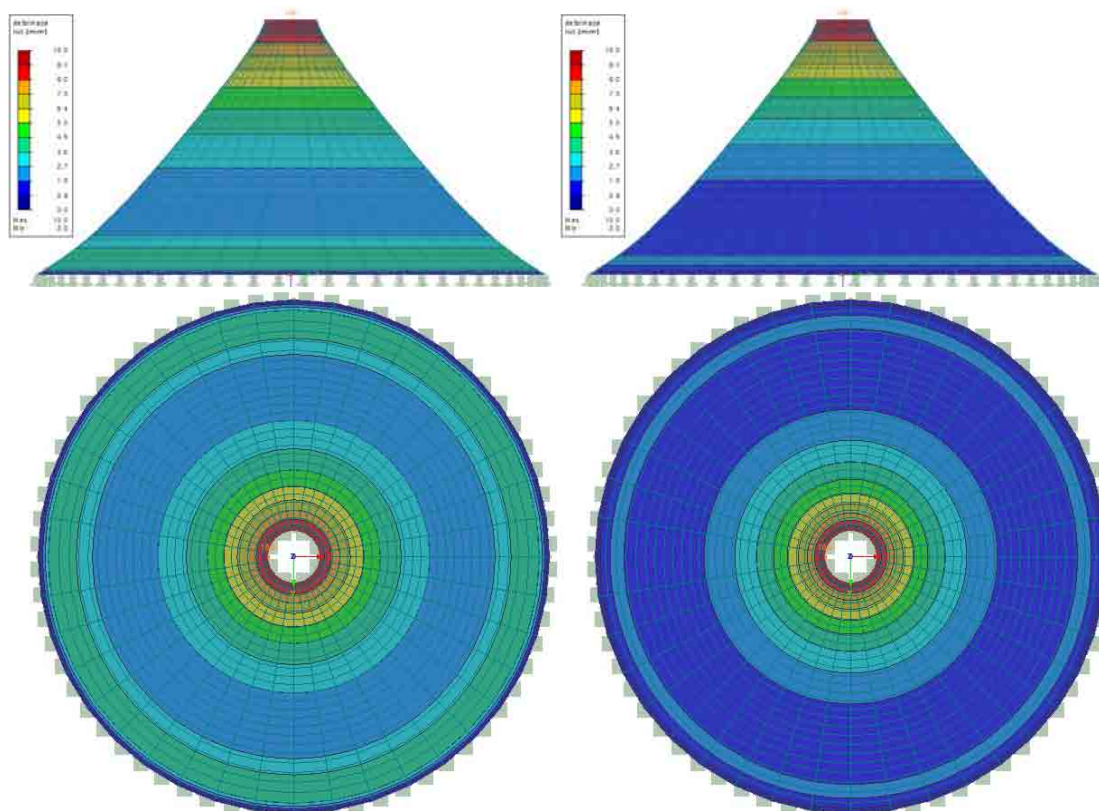


Obr. 55 Řezy tvarů pro jednotlivá poměrná přetvoření

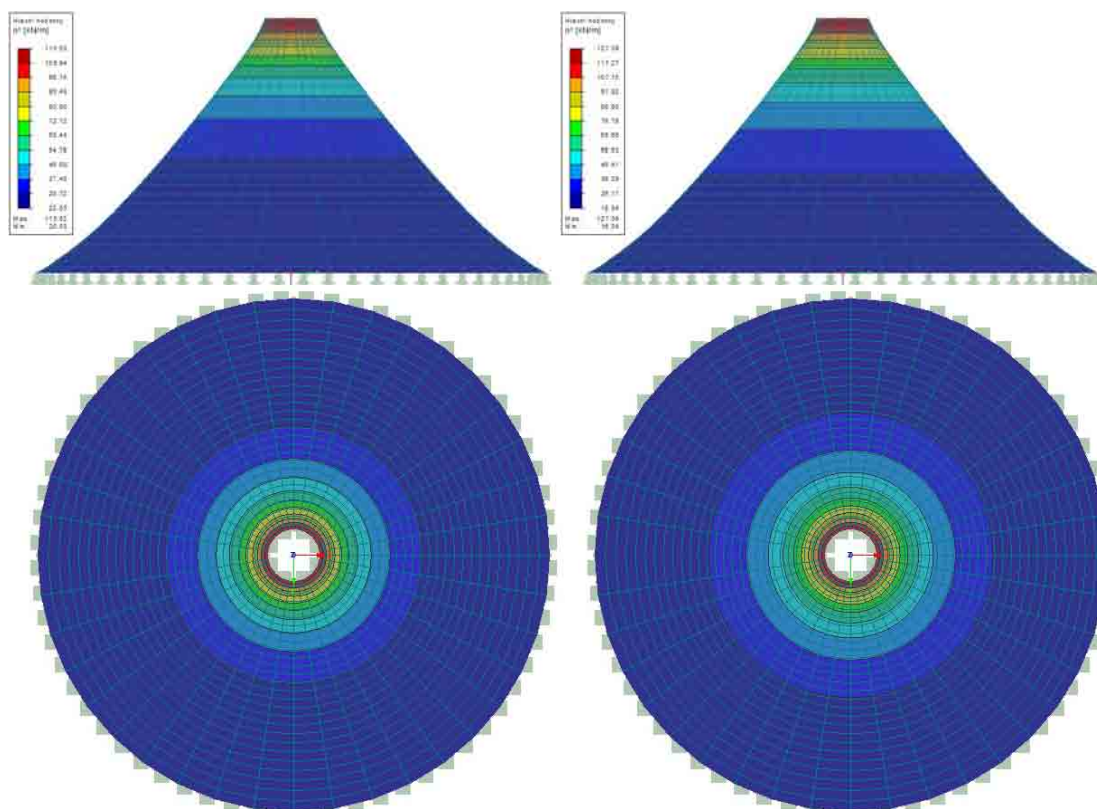
Jak je patrné z obr. 55, tak při 20% přetvoření vzniká tvar s nežádoucím zúžením u vrcholu a proto již tento tvar nebudu analyzovat.

3.3.2.1.1.1 Tvar 1

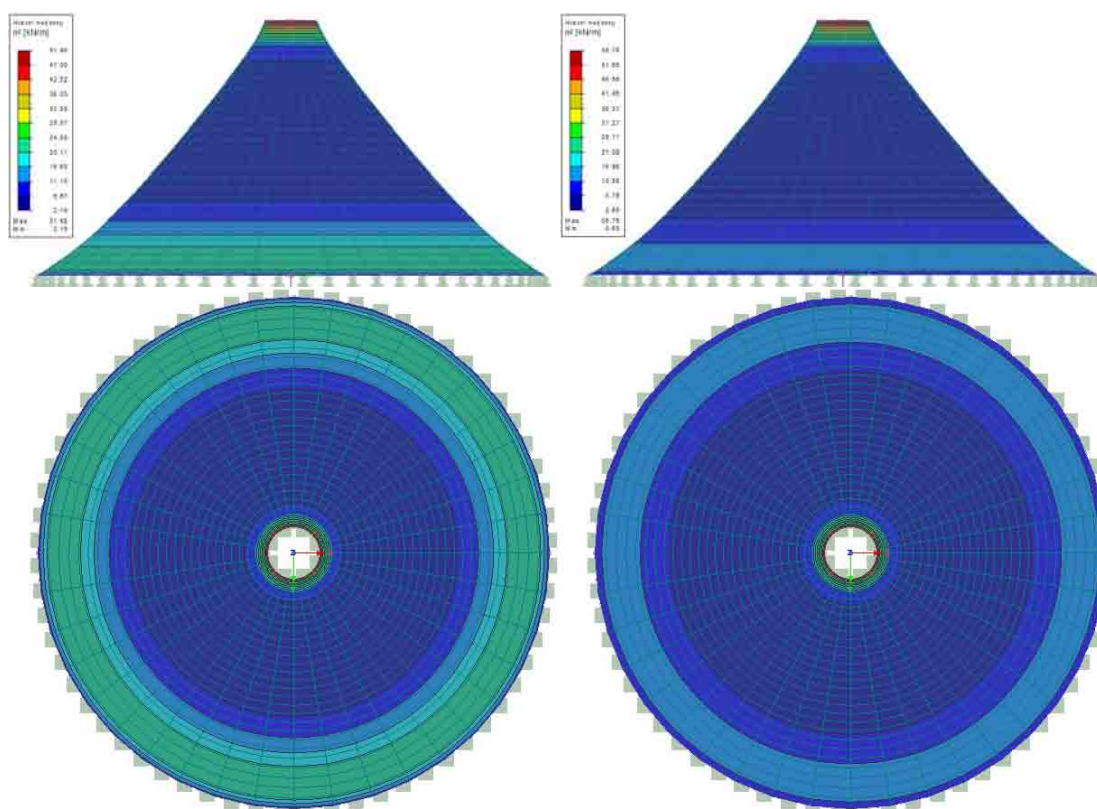
Na prvním tvaru vznikly deformace při stálém zatížení velikosti max. 10 mm, což odpovídá deformaci vnitřní liniové podpory. Při zatížení sněhem nepřesáhla maximální deformace hodnotu 10 mm. Hlavní napětí n_1 vzrostlo ze 115.52 kN/m pro stálé zatížení na 127.39 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z 2.19 kN/m pro stálé zatížení na 0.69 kN/m pro kombinaci se sněhem. V tomto tvaru tedy nevznikly žádné tlaky ani v jedné kombinaci a membrána je pouze tažena. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 56 Deformace při stálém zatížení a kombinaci stálého se sněhem



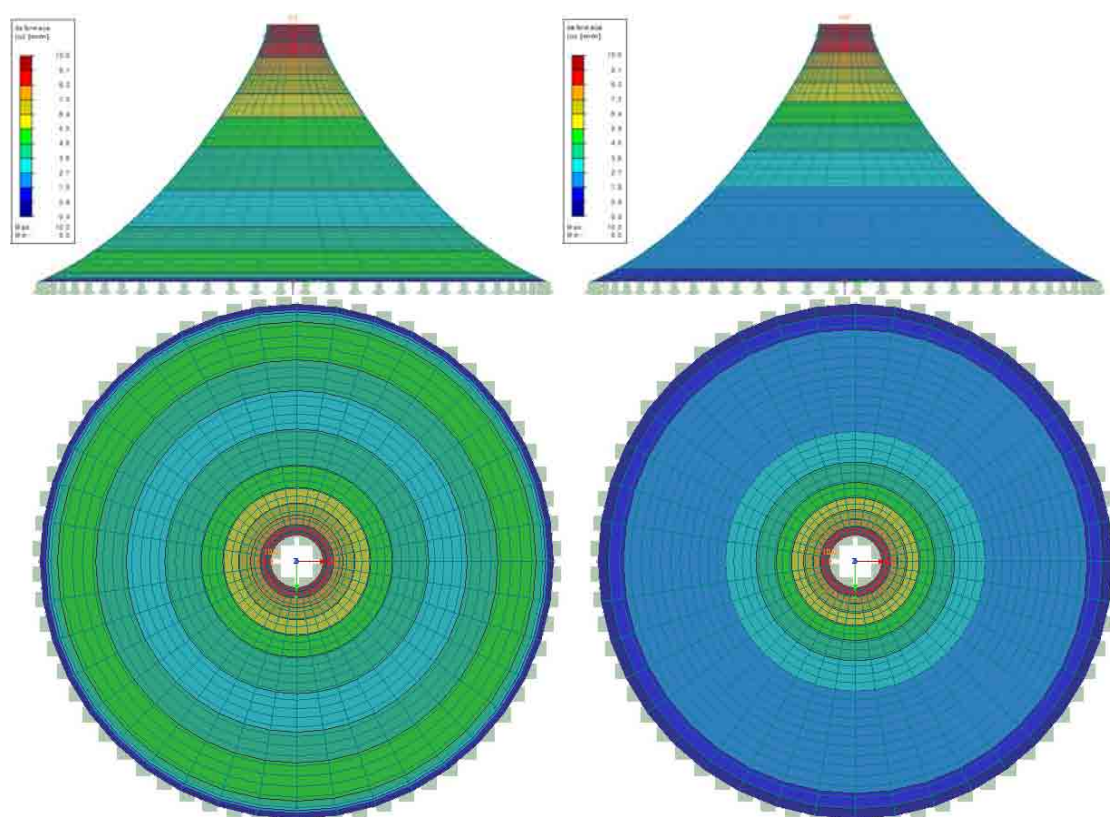
Obr. 57 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



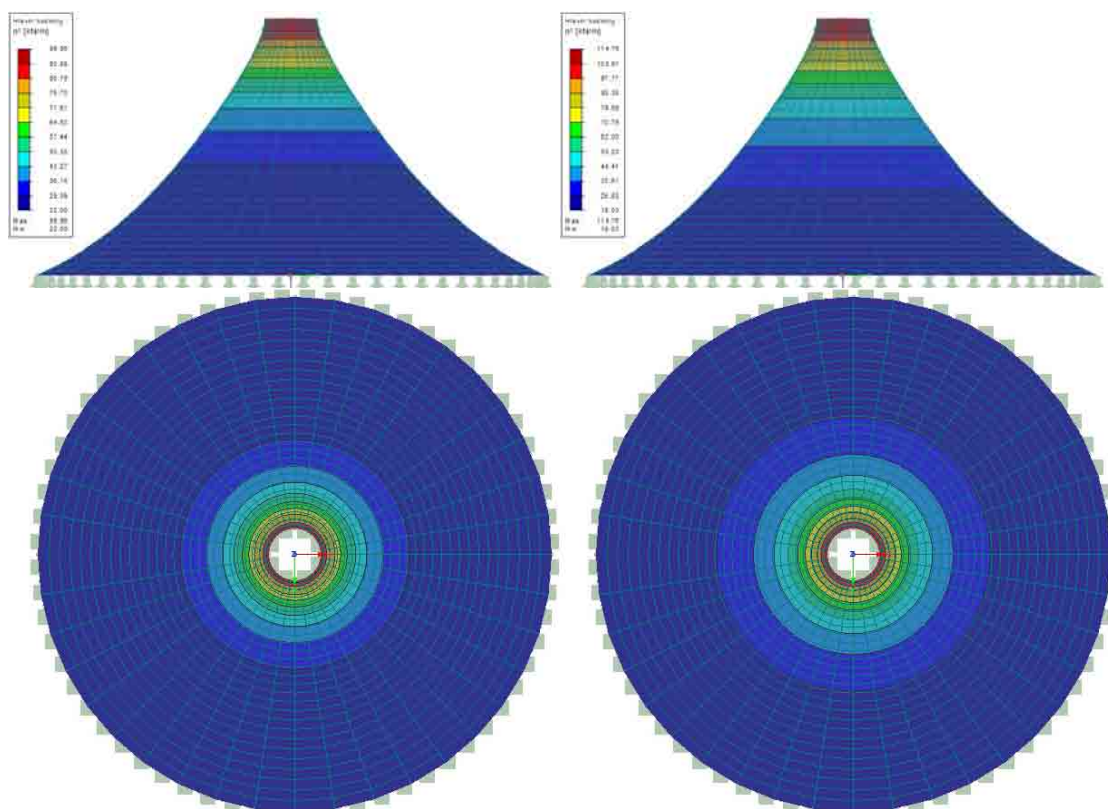
Obr. 58 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.1.1.2 Tvar 2

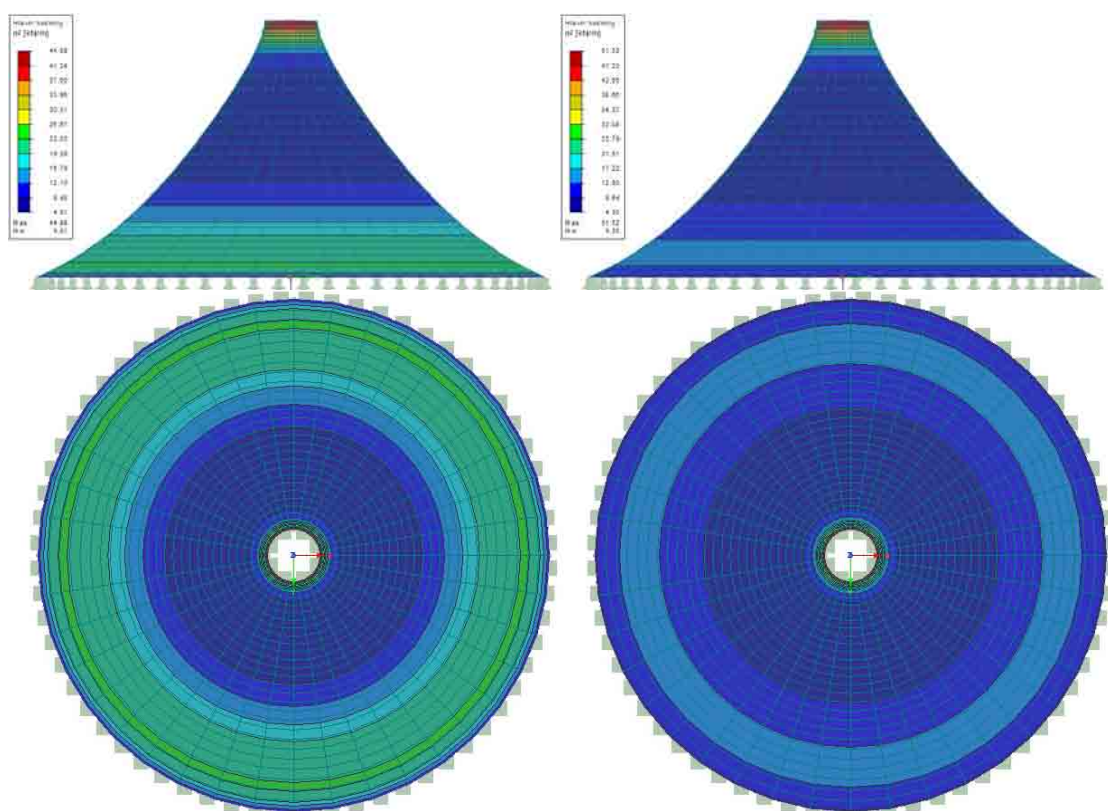
Na druhém tvaru opět deformace nepřekročily hodnotu danou vynuceným posunem podpor. Hlavní napětí n_1 vzrostlo ze 99.96 kN/m pro stálé zatížení na 114.76 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z 4.61 kN/m pro stálé zatížení na 4.35 kN/m pro kombinaci se sněhem. Membrána je pro tento tvar opět pouze tažena. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 59 Deformace při stálém zatížení a kombinaci stálého se sněhem



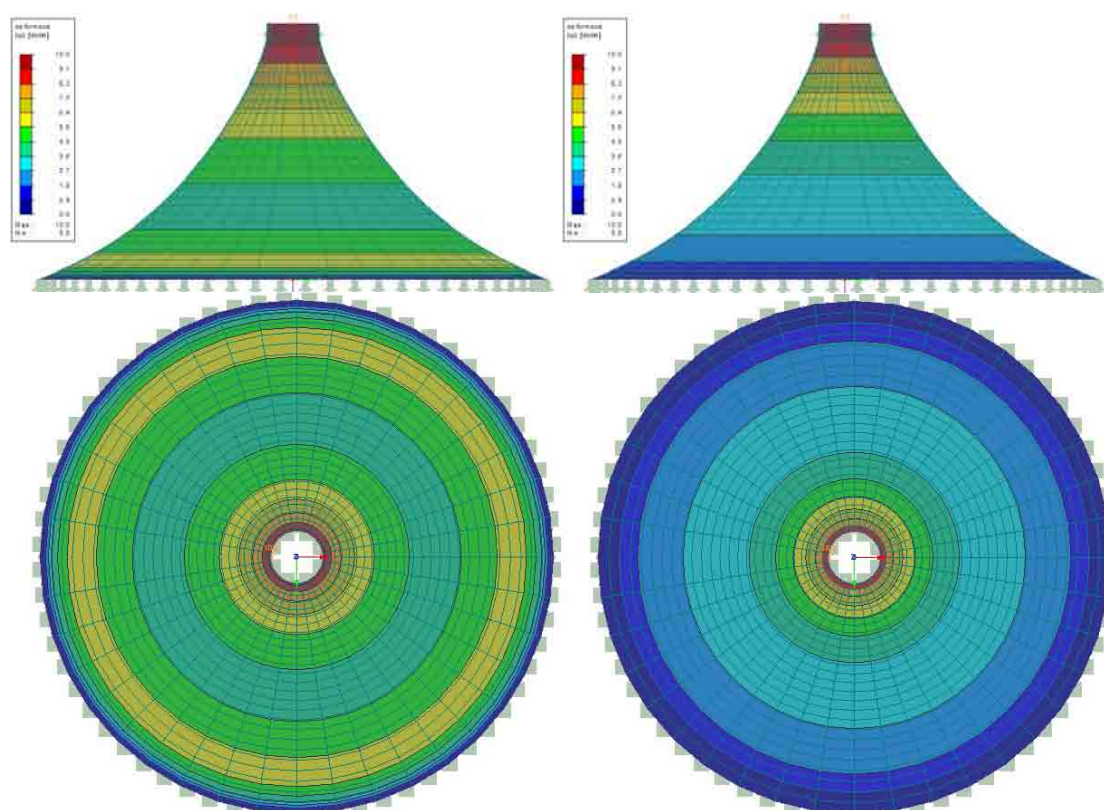
Obr. 60 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



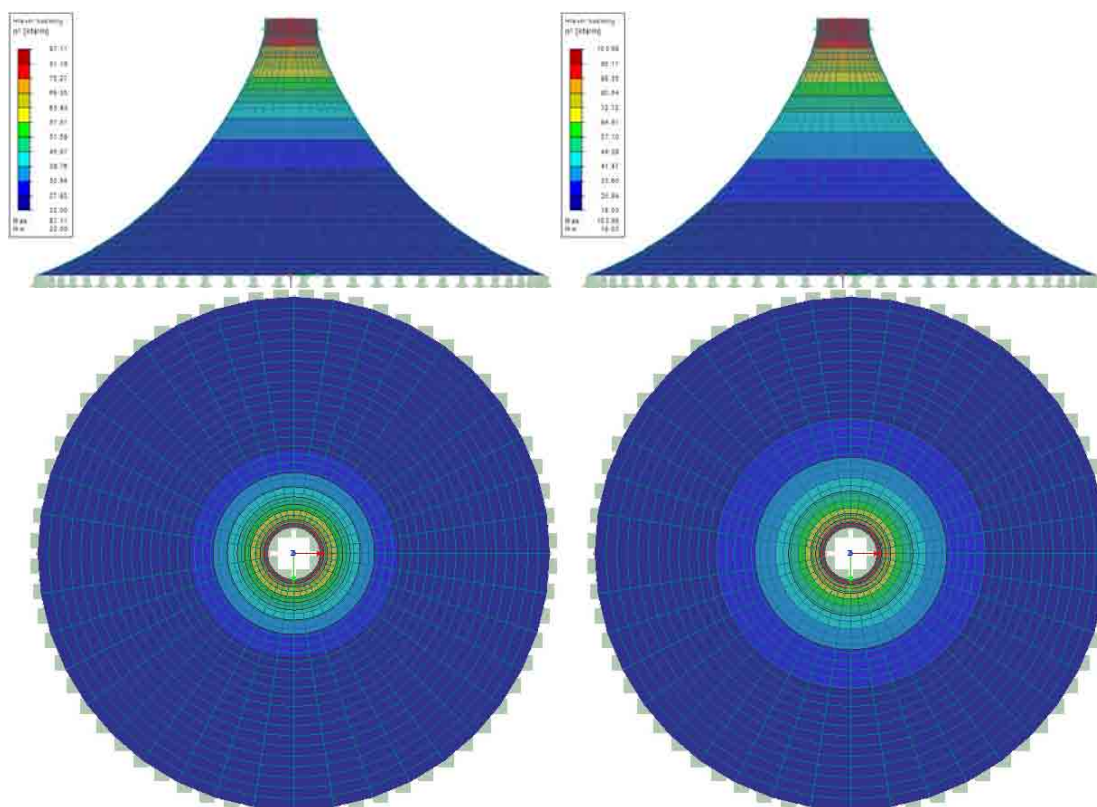
Obr. 61 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.1.1.3 Tvar 3

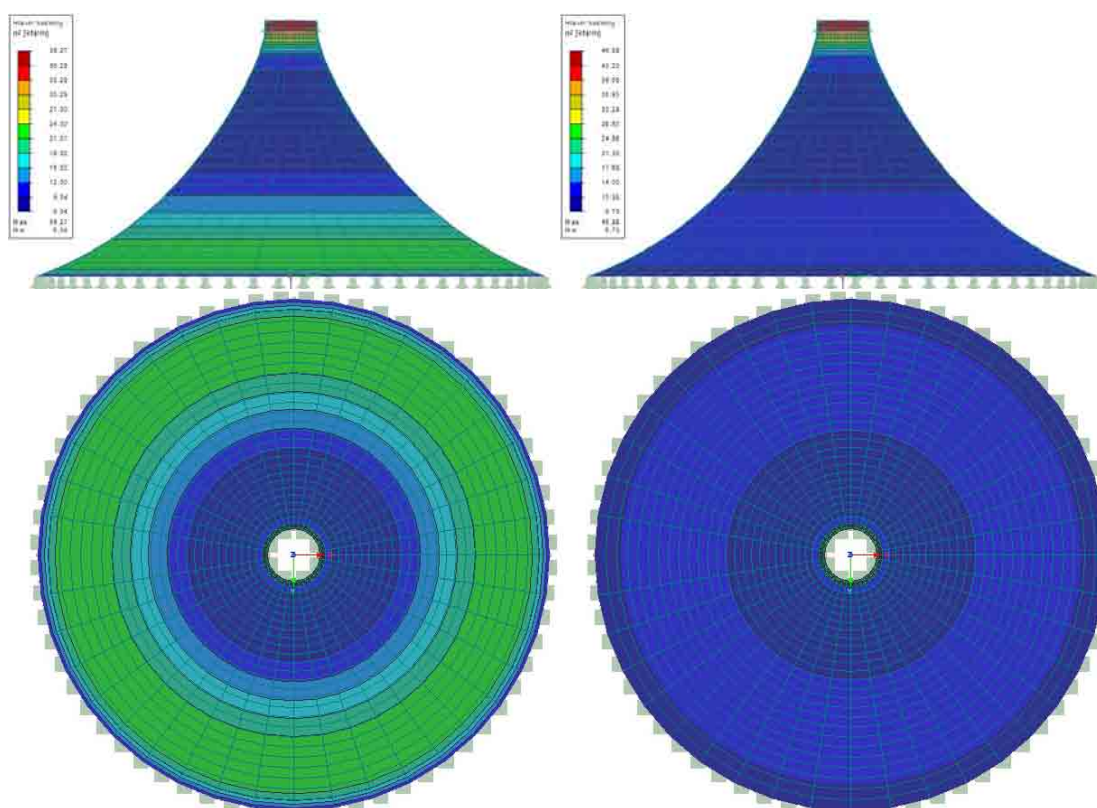
Ani na třetím tvaru deformace nepřesáhly hodnotu 10 mm. Hlavní napětí n_1 vzrostlo ze 87.11 kN/m pro stálé zatížení na 103.98 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 vzrostly z 6.34 kN/m pro stálé zatížení na - 6.73 kN/m pro kombinaci se sněhem. Membrána je opět celá tažena. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 62 Deformace při stálém zatížení a kombinaci stálého se sněhem



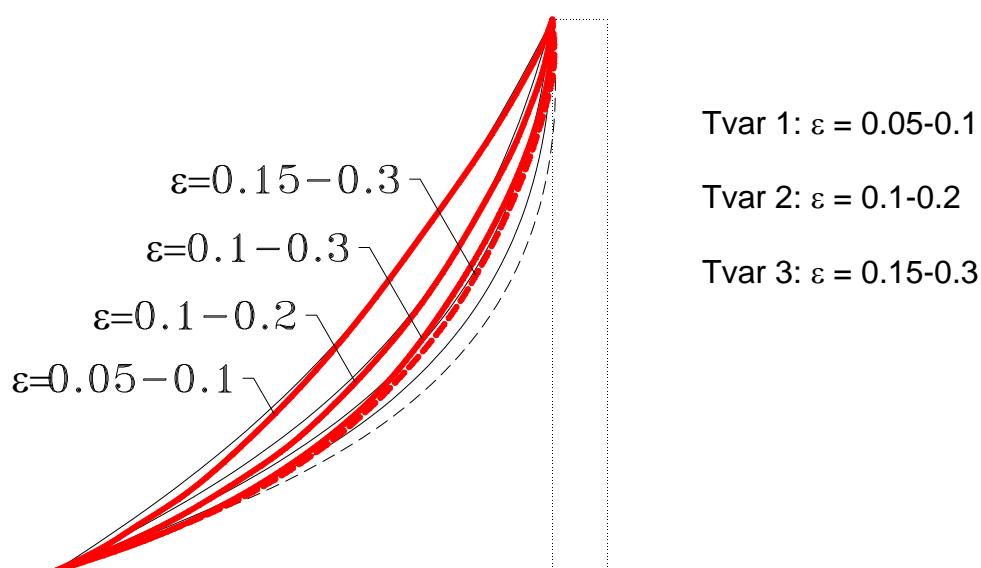
Obr. 63 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



Obr. 64 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.1.2 Poměrné přetvoření s lineární změnou ve směru osy z

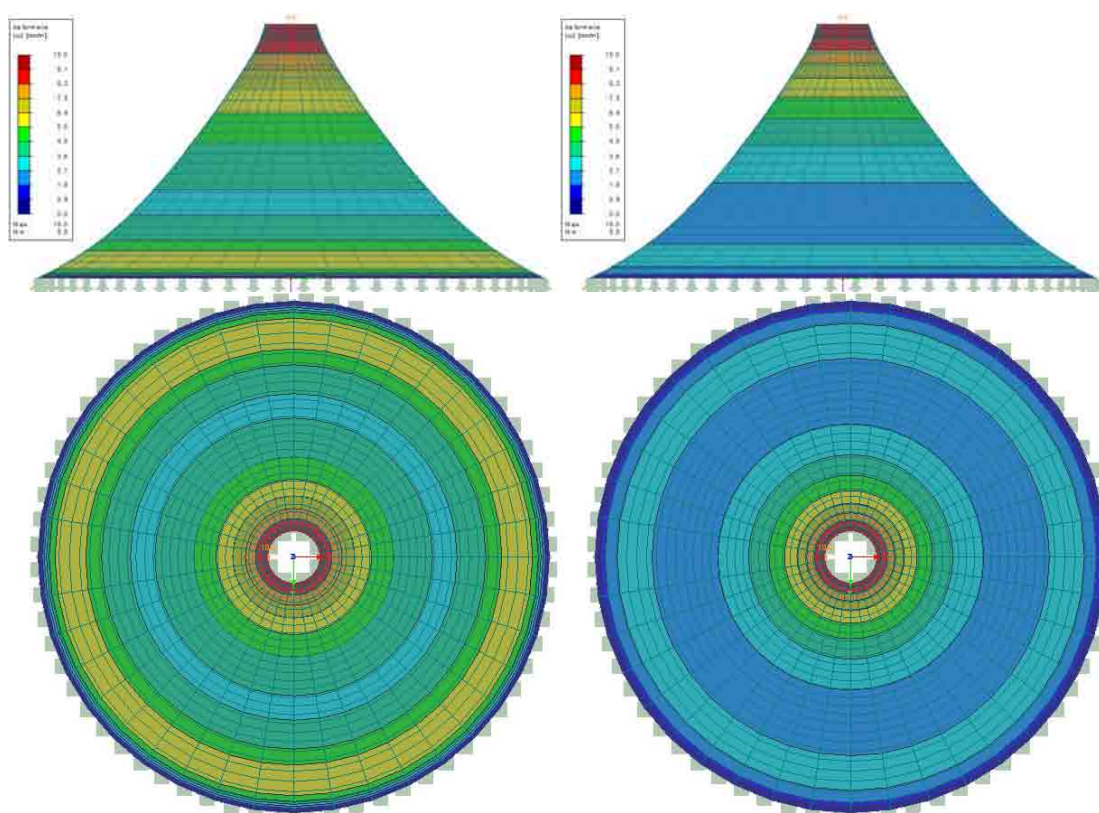
Vzhledem k tomu, že spodní část rotační plochy vykazuje větší deformační kapacitu, zvolil jsem zatížení poměrným přetvořením s lineární změnou ve směru osy z, které RFEM také podporuje. Volil jsem nárůst na dvojnásobek poměrného přetvoření u vrcholu a to přetvoření o 5-10%, 10-20% a 15-30%. Vzhledem k nevhodnosti tvaru s největším poměrným přetvořením jsem jej nahradil přetvořením o 10-30%.



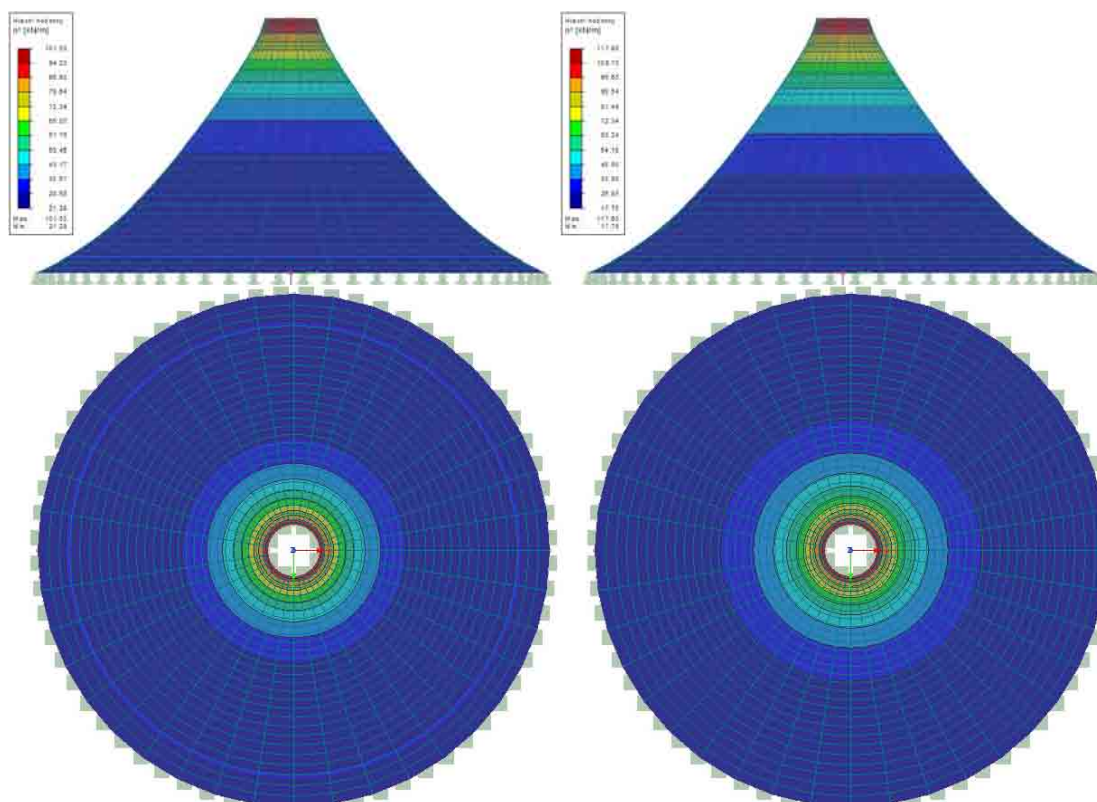
Obr. 65 Řezy tvarů pro jednotlivá poměrná přetvoření

3.3.2.1.2.1 Tvar 1

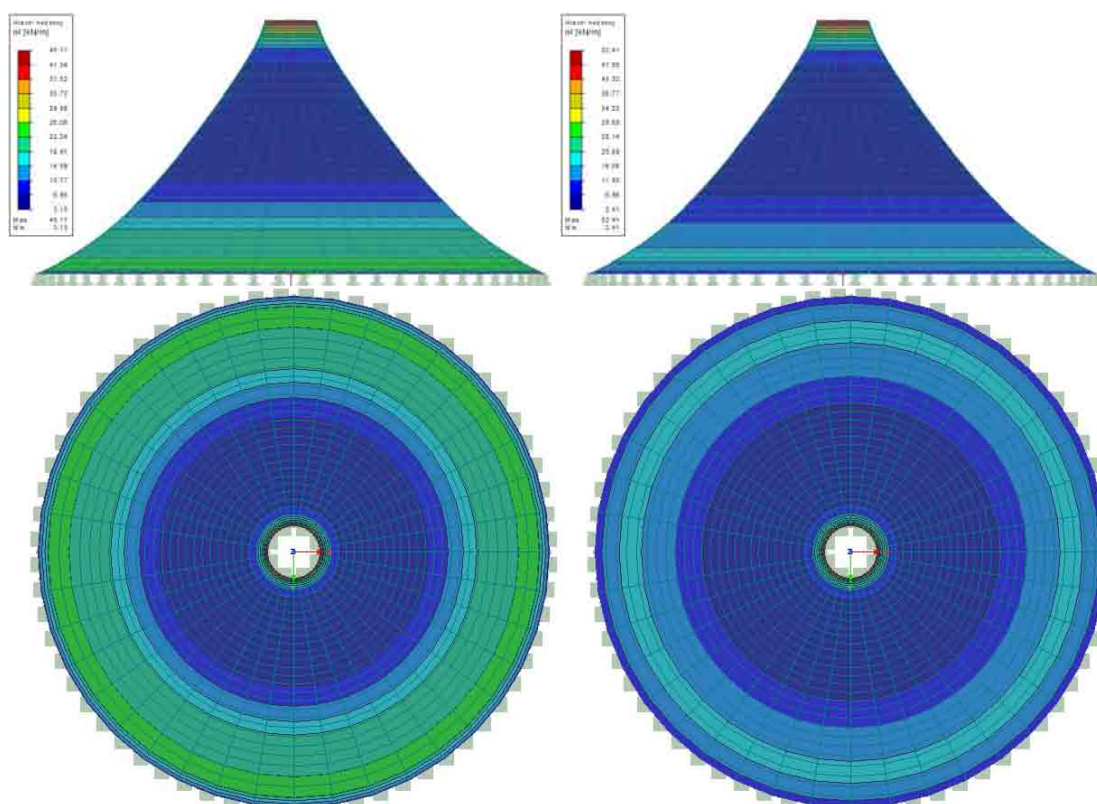
Na prvním tvaru vznikly deformace při stálém zatížení velikosti max. 10 mm, což odpovídá deformaci vnitřní liniové podpory. Při zatížení sněhem zůstala maximální deformace na 10 mm. Hlavní napětí n_1 vzrostlo ze 101.52 kN/m pro stálé zatížení na 117.83 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z 3.13 kN/m pro stálé zatížení na 2.41 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 66 Deformace při stálém zatížení a kombinaci stálého se sněhem



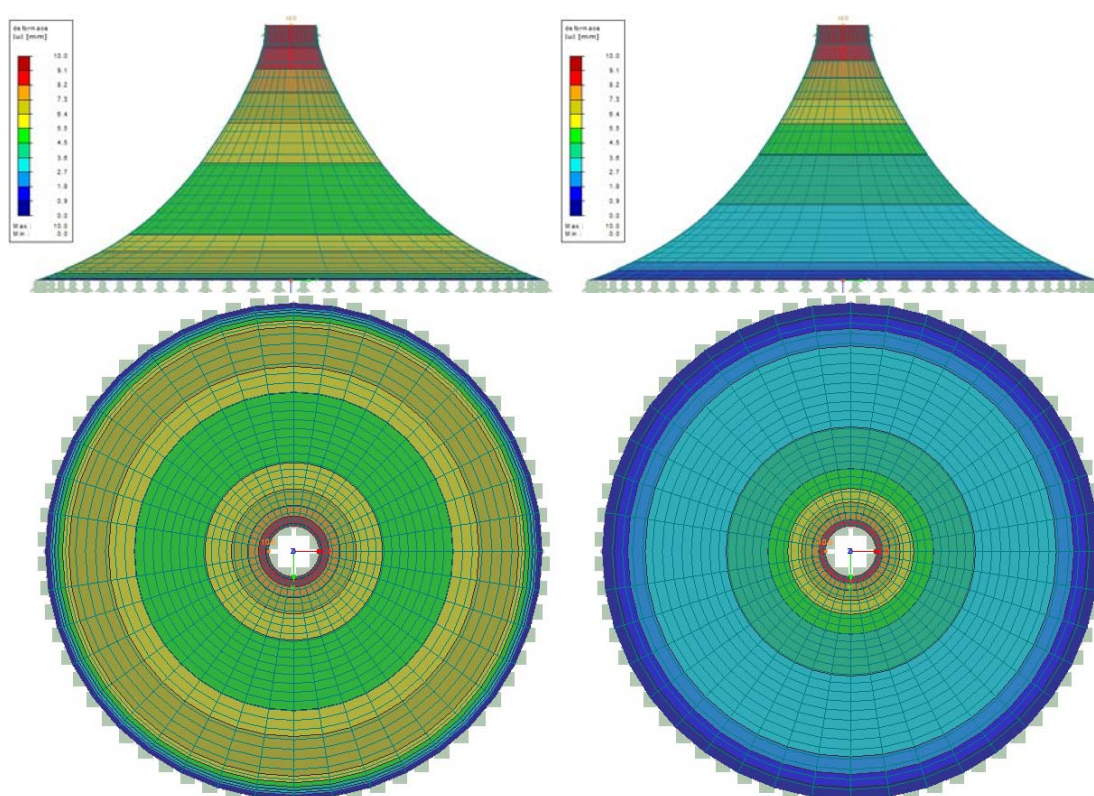
Obr. 67 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



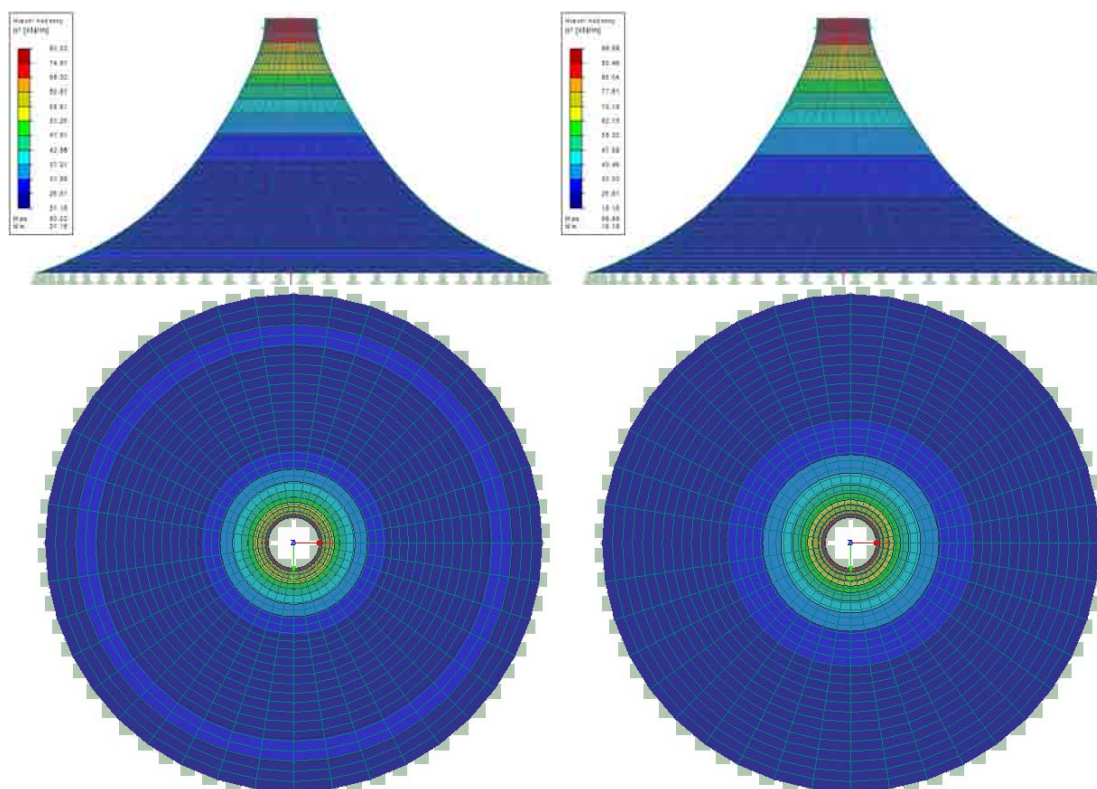
Obr. 68 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.1.2.2 Tvar 2

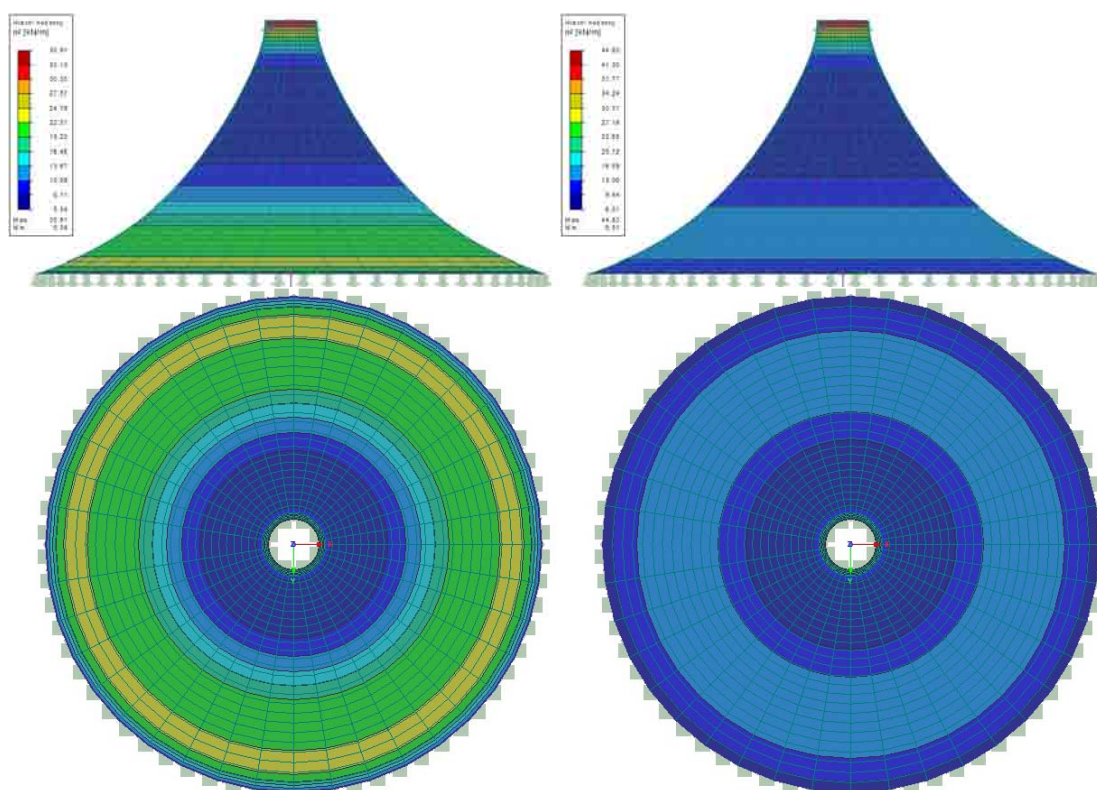
Na druhém tvaru vznikly opět deformace o velikosti max. 10 mm jak při stálém zatížení, tak i při kombinaci se sněhem. Hlavní napětí n_1 vzrostlo ze 80.02 kN/m pro stálé zatížení na 99.89 kN/m v kombinaci se sněhem. Nejvyšší hodnoty hlavního napětí n_2 klesly z 5.34 kN/m pro stálé zatížení na 5.01 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 69 Deformace při stálém zatížení a kombinaci stálého se sněhem



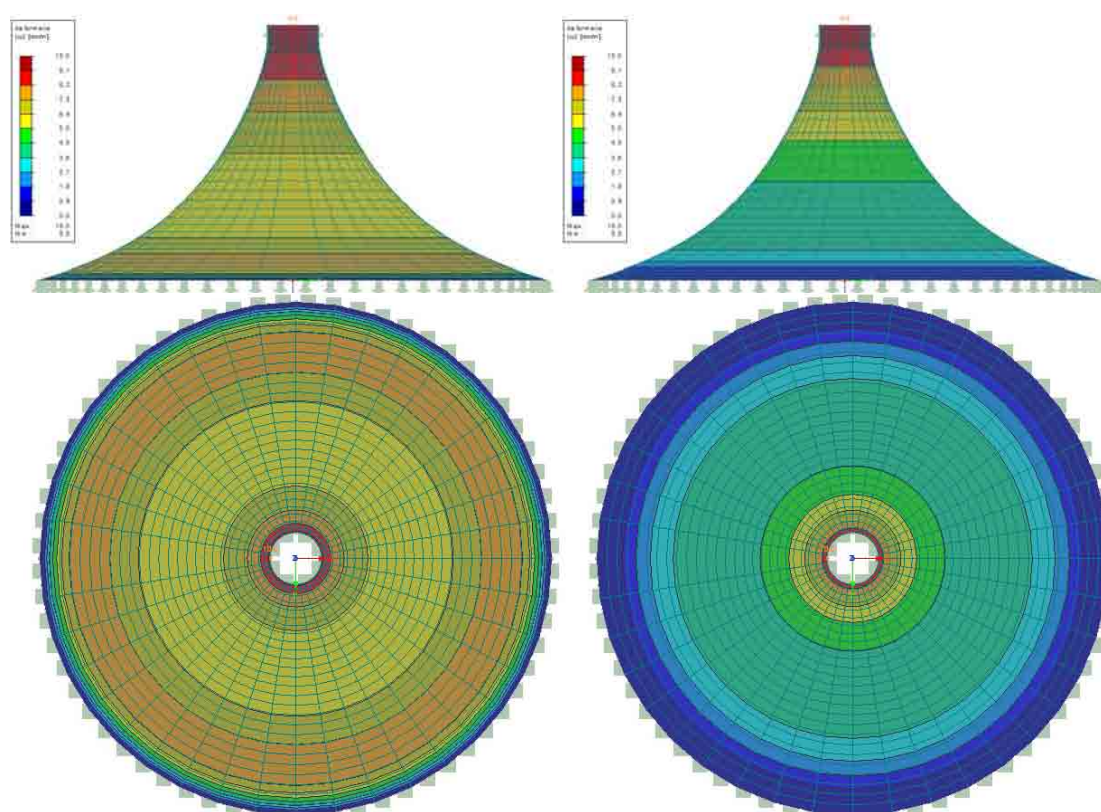
Obr. 70 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



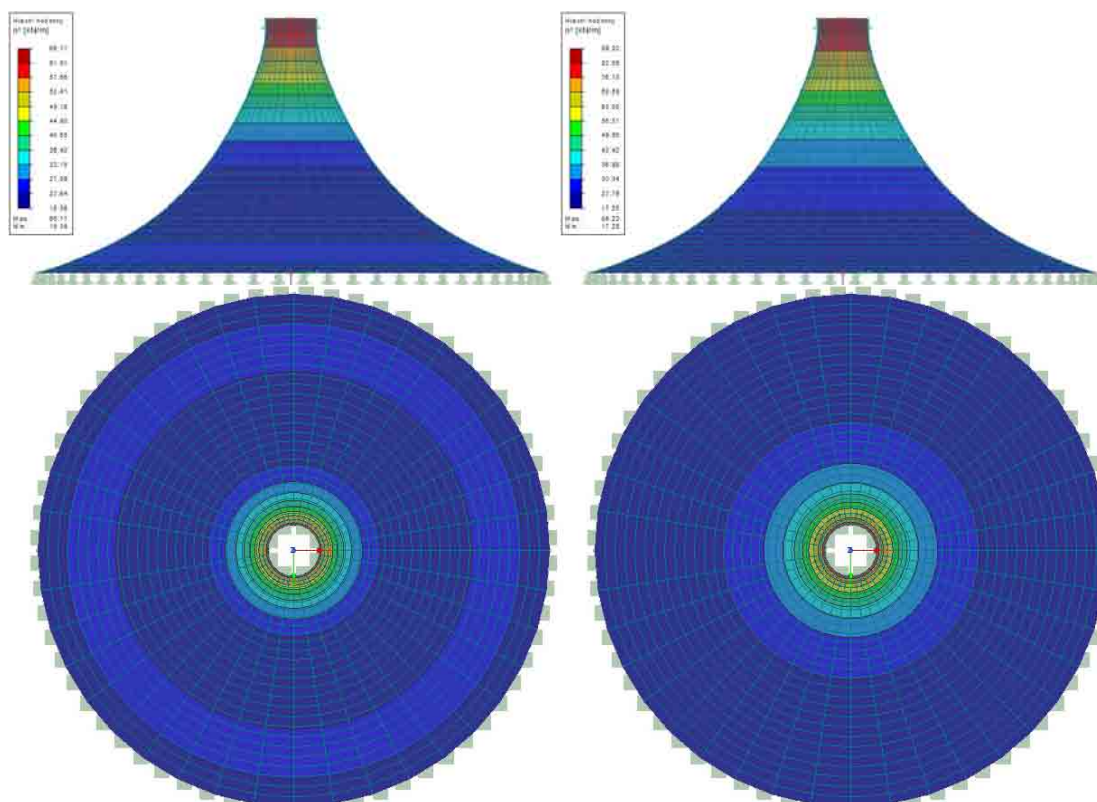
Obr. 71 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.1.2.3 Tvar 3

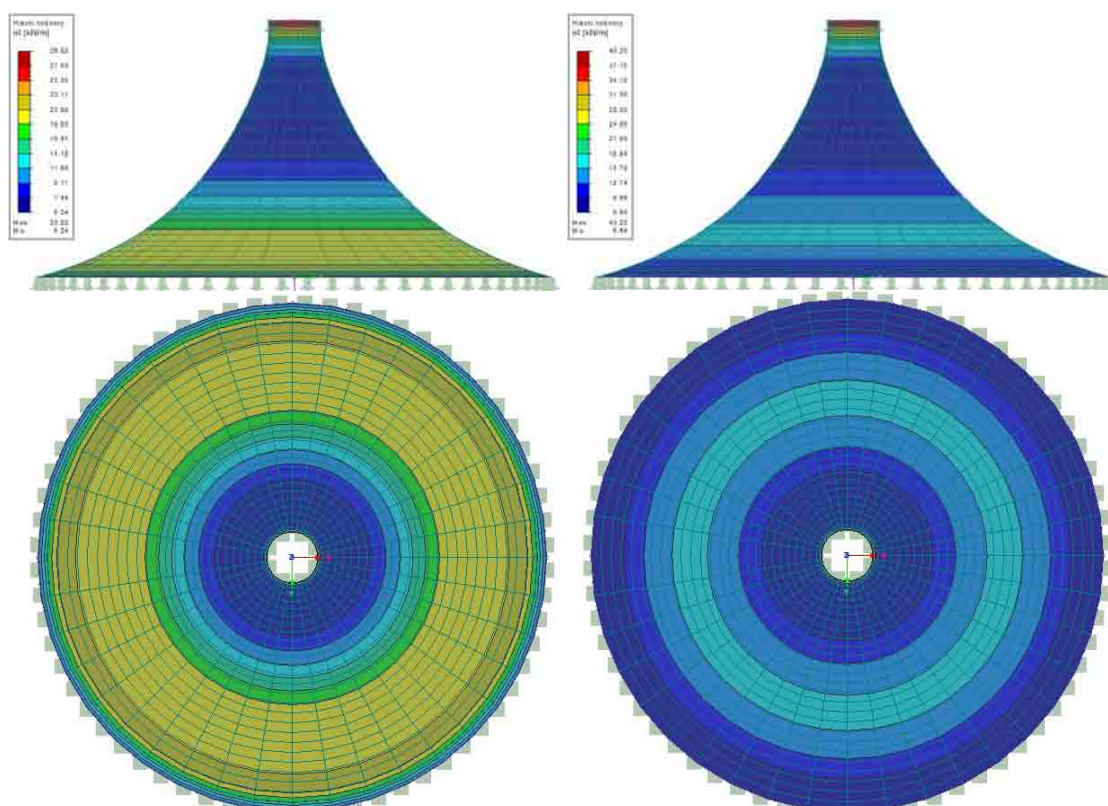
Na druhém tvaru vznikly opět deformace o velikosti max. 10 mm jak při stálém zatížení, tak i při kombinaci se sněhem. Hlavní napětí n_1 vzrostlo ze 66.17 kN/m pro stálé zatížení na 89.22 kN/m v kombinaci se sněhem. Nejvyšší hodnoty hlavního napětí n_2 vzrostly z 5.24 kN/m pro stálé zatížení na 6.64 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 72 Deformace při stálém zatížení a kombinaci stálého se sněhem



Obr. 73 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



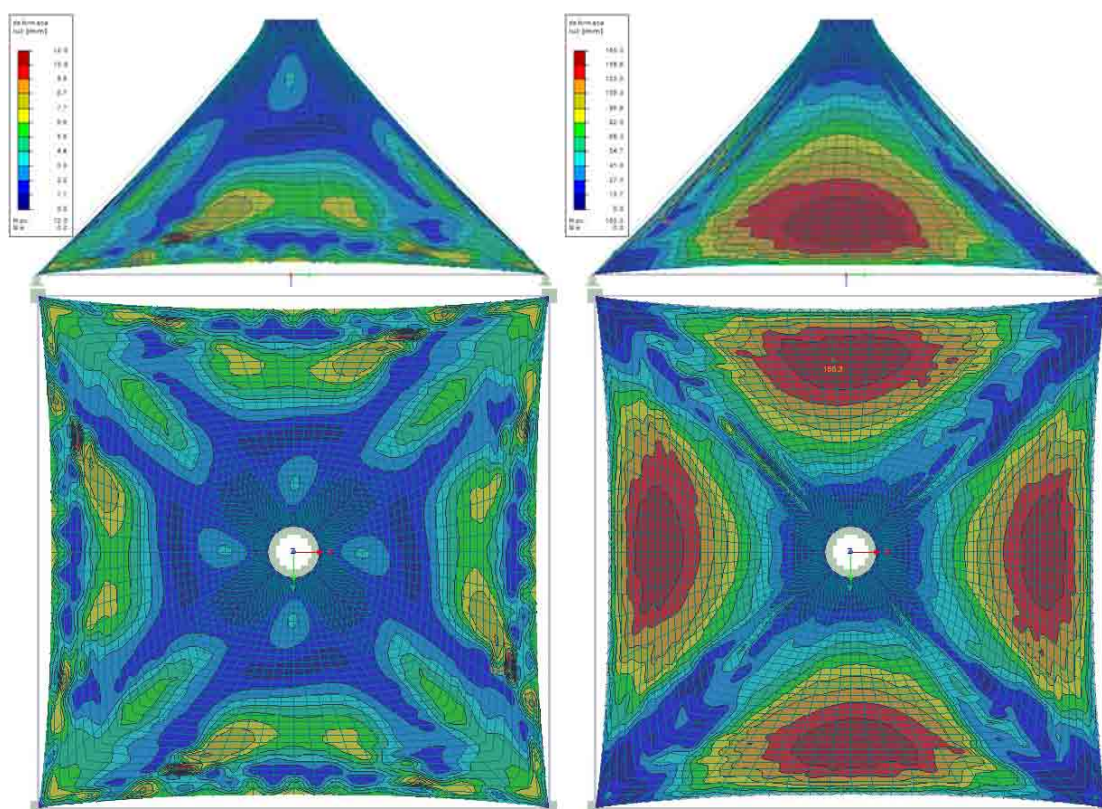
Obr. 74 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.2 Čtvercový půdorys

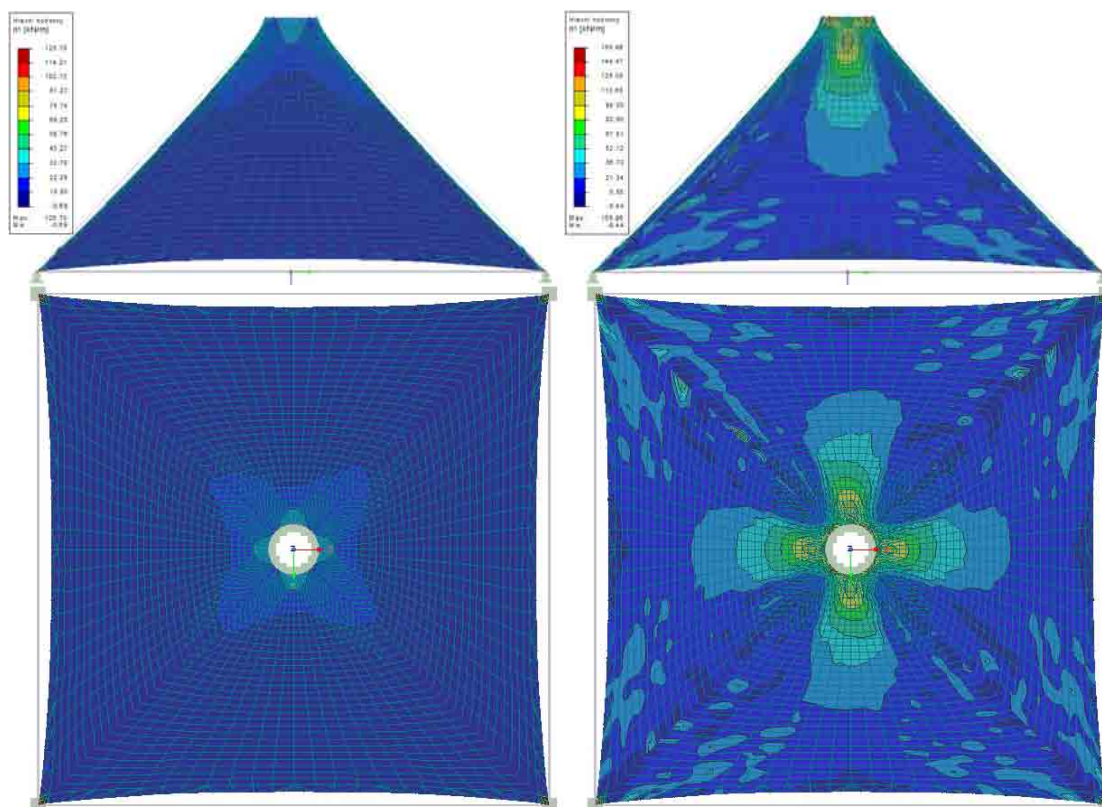
3.3.2.2.1 Konstantní poměrné přetvoření na celé ploše

3.3.2.2.1.1 Tvar 1

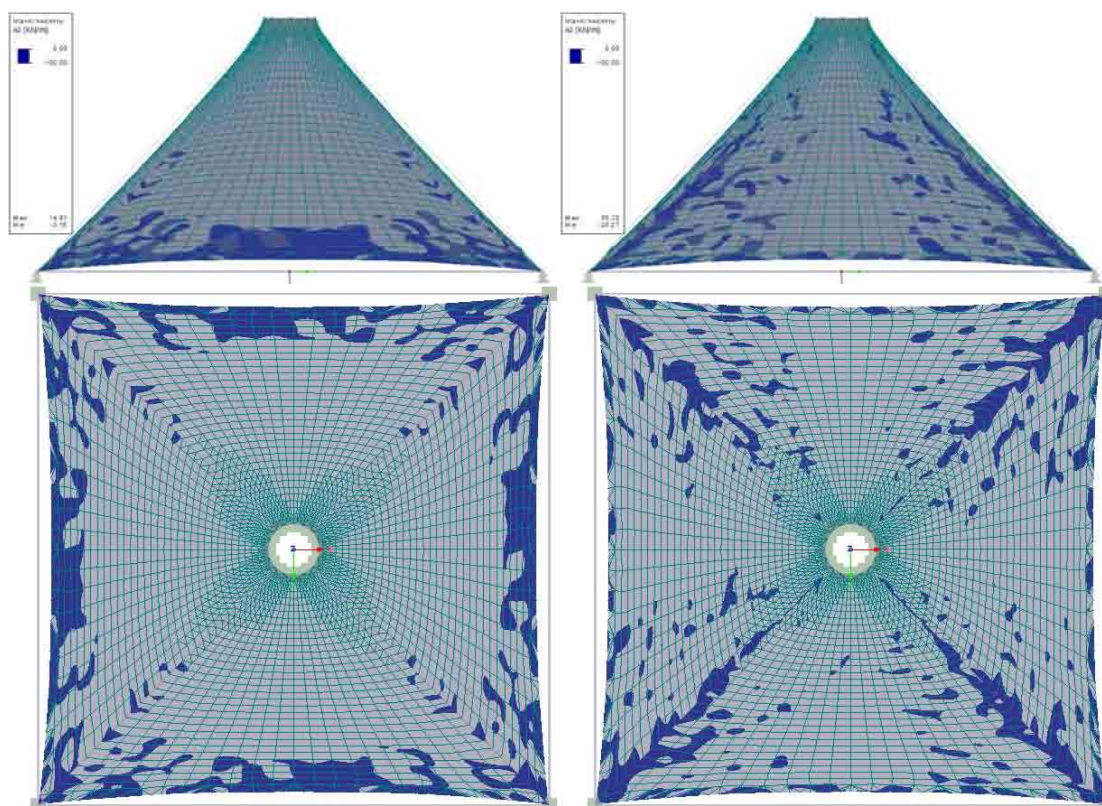
Na prvním tvaru vznikly deformace o velikosti max. 12.0 mm při stálém zatížení a při kombinaci se sněhem o velikosti 150.3 mm. Hlavní napětí n_1 vzrostlo ze 125.70 kN/m pro stálé zatížení na 159.86 kN/m v kombinaci se sněhem. V tomto případě bohužel opět nabývá hlavní napětí n_1 záporných hodnot a to -0.69 kN/m pro stálé zatížení a -9.44 kN/m pro kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -3.16 kN/m pro stálé zatížení na -28.27 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 75 Deformace při stálém zatížení a kombinaci stálého se sněhem



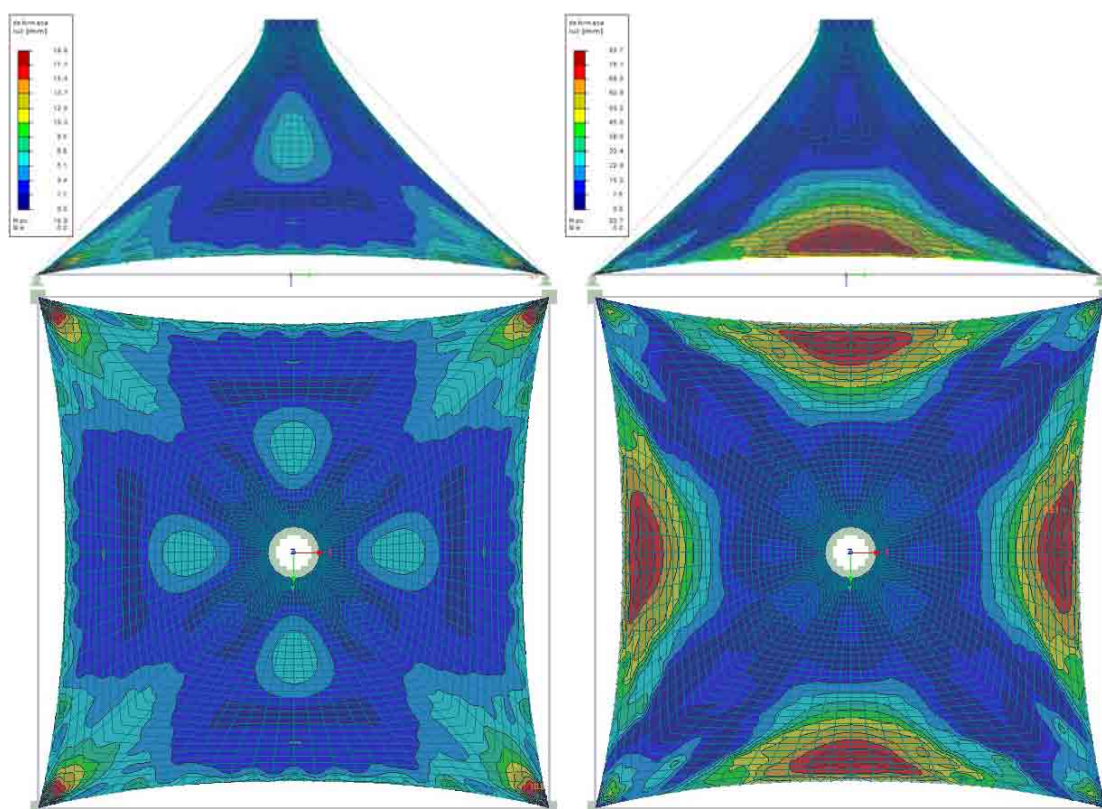
Obr. 76 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



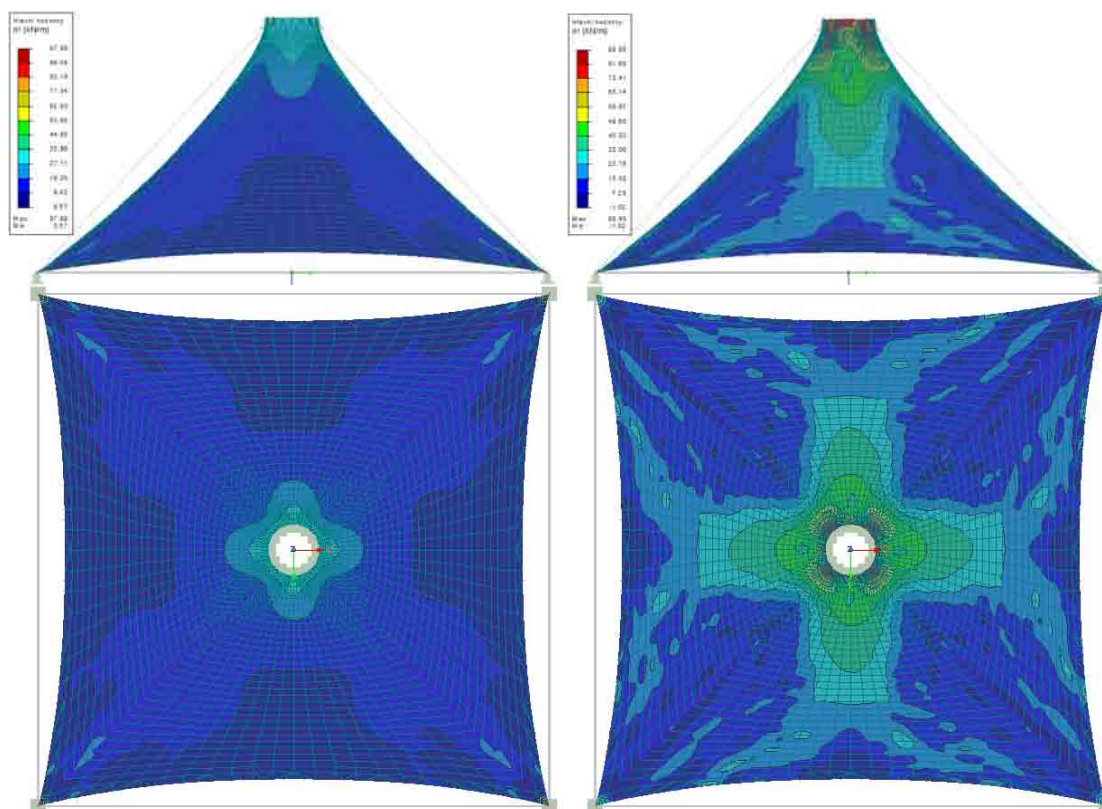
Obr. 77 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.2.1.2 Tvar 2

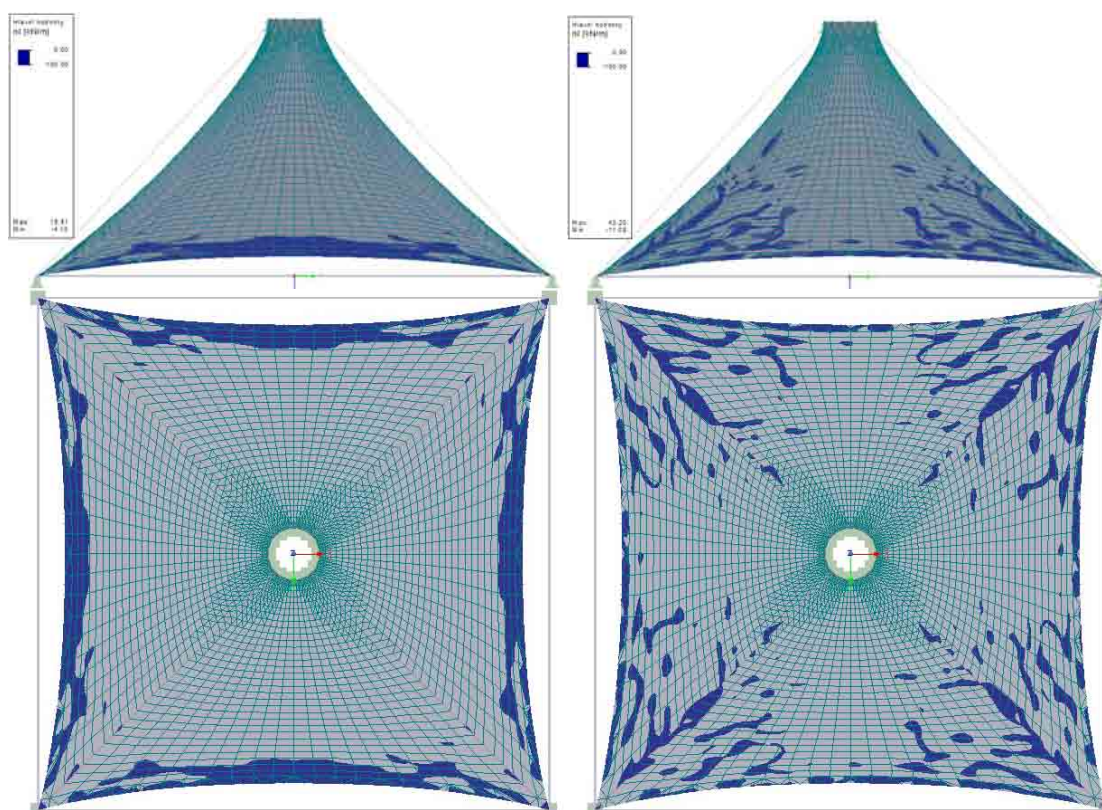
Na druhém tvaru vznikly deformace o velikosti max. 18.8 mm při stálém zatížení a při kombinaci se sněhem o velikosti 83.7 mm. Hlavní napětí n_1 kleslo ze 97.88 kN/m pro stálé zatížení na 89.95 kN/m v kombinaci se sněhem. V případě kombinace se sněhem nabývají hodnoty hlavního napětí n_1 záporných hodnot a to -1.02 kN/m. Nejnižší hodnoty hlavního napětí n_2 klesly z -4.13 kN/m pro stálé zatížení na -11.08 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 78 Deformace při stálém zatížení a kombinaci stálého se sněhem



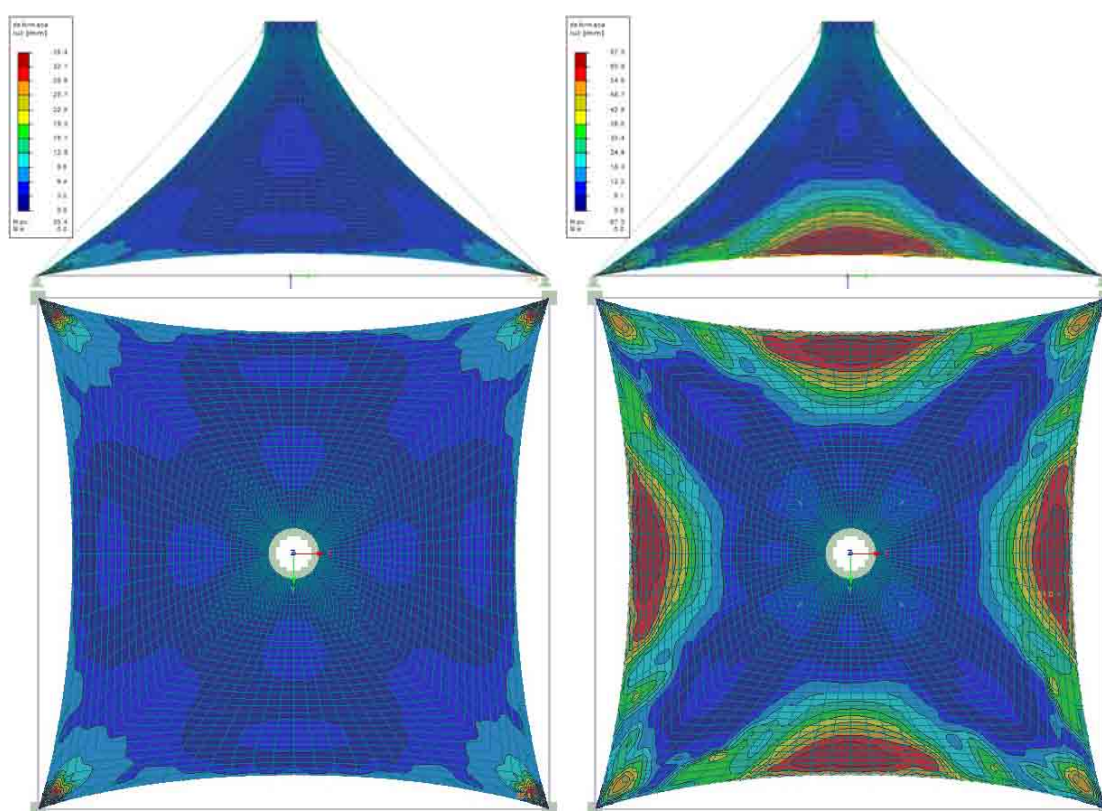
Obr. 79 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



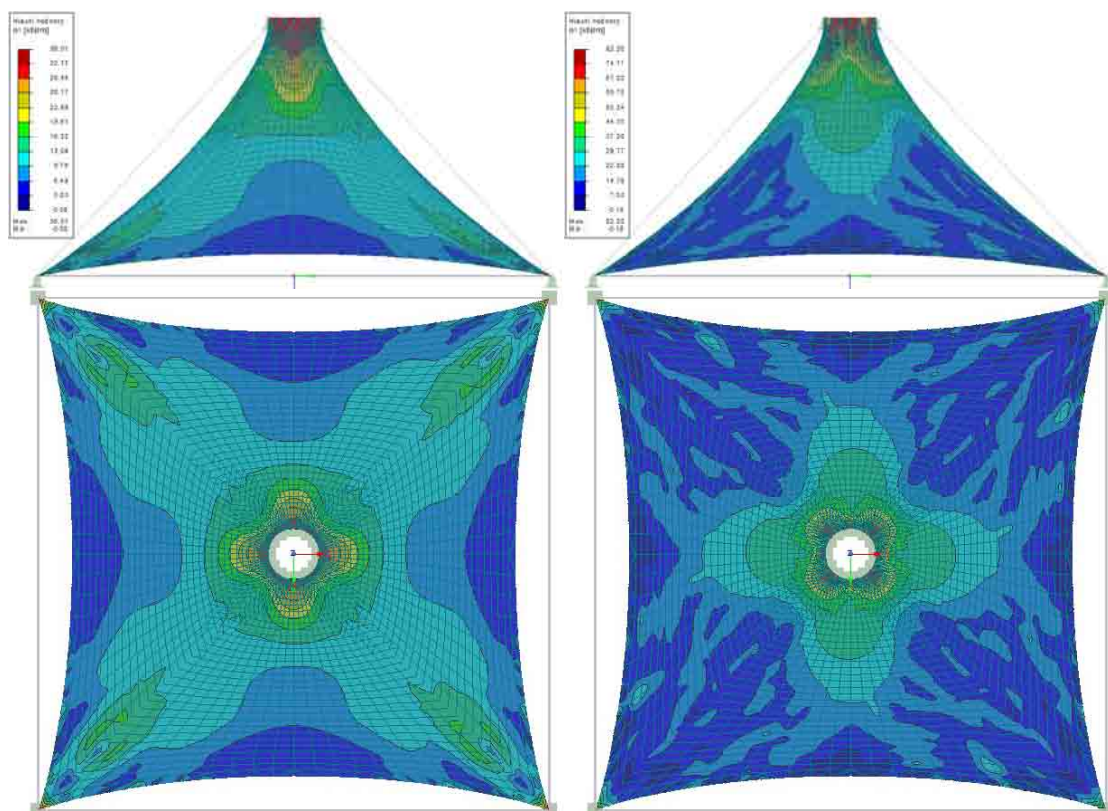
Obr. 80 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.2.1.3 Tvar 3

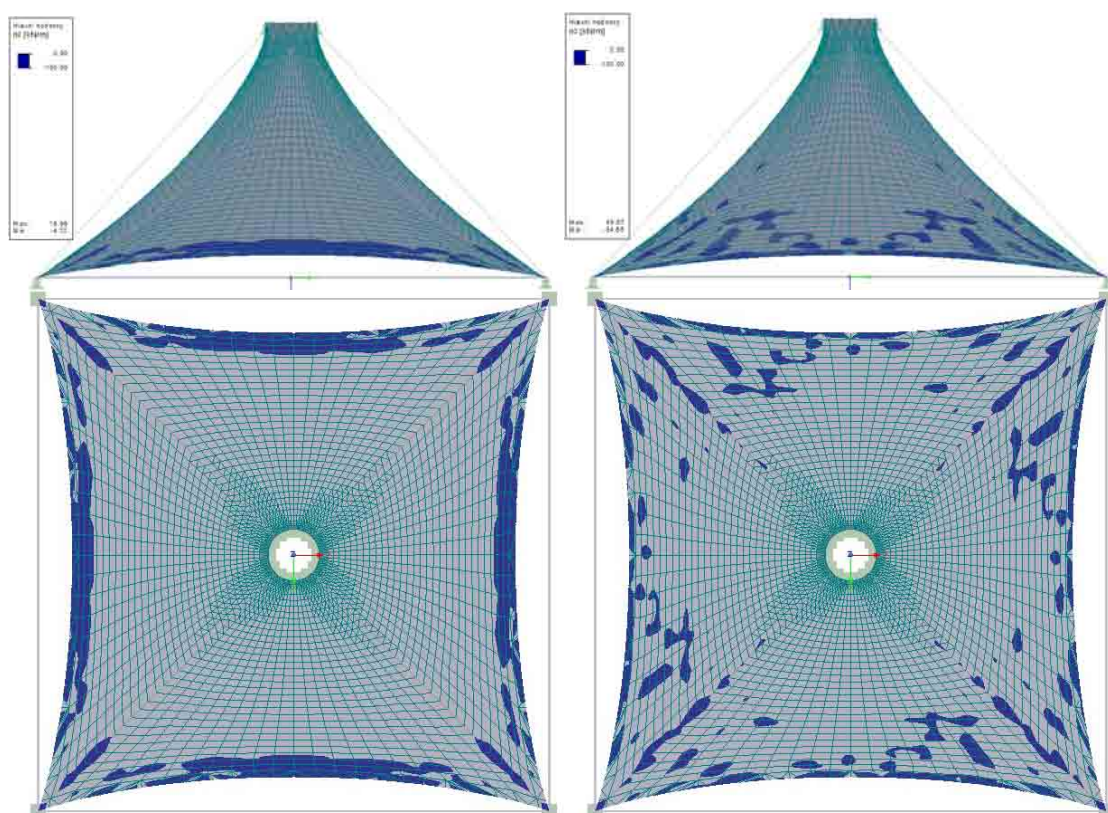
Na třetím tvaru vznikly deformace o velikosti max. 35.4 mm při stálém zatížení a při kombinaci se sněhem o velikosti 67.0 mm. Hlavní napětí n_1 vzrostlo ze 36.01 kN/m pro stálé zatížení na 82.20 kN/m v kombinaci se sněhem. V tomto případě bohužel opět nabývá hlavní napětí n_1 záporných hodnot a to -0.08 kN/m pro stálé zatížení a -0.19 kN/m pro kombinaci se sněhem. Nejnížší hodnoty hlavního napětí n_2 klesly z -4.72 kN/m pro stálé zatížení na -34.65 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 81 Deformace při stálém zatížení a kombinaci stálého se sněhem



Obr. 82 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem

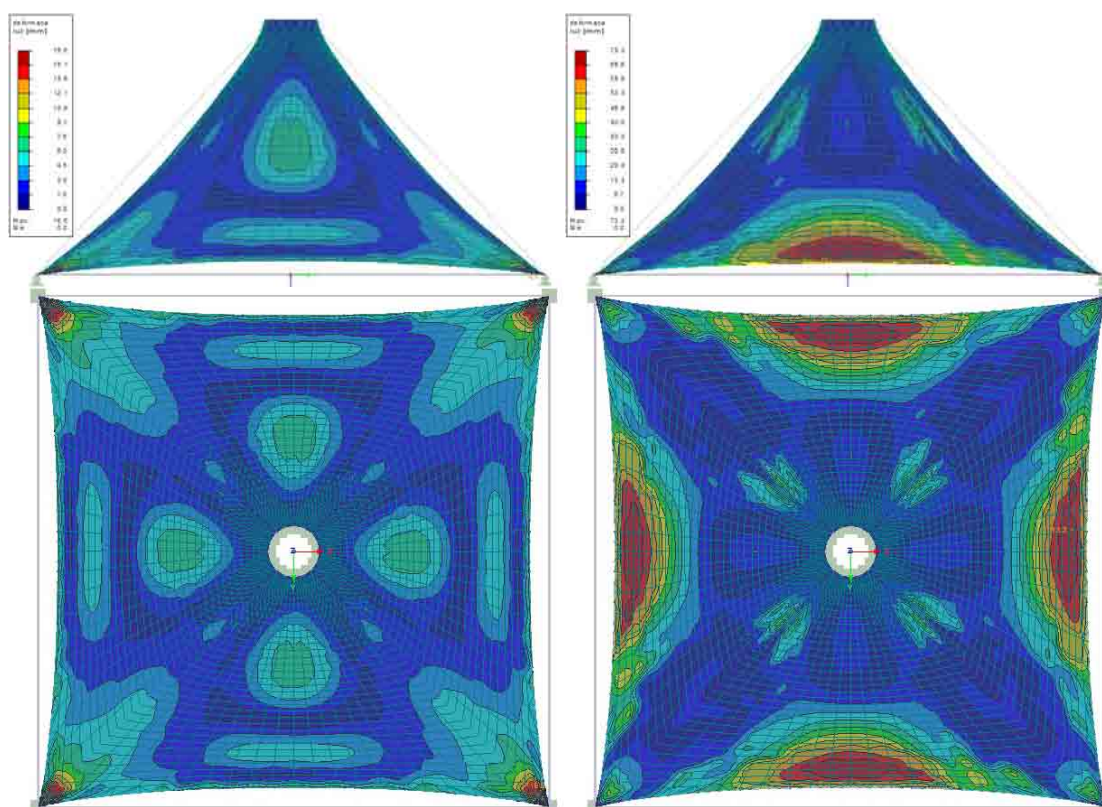


Obr. 83 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

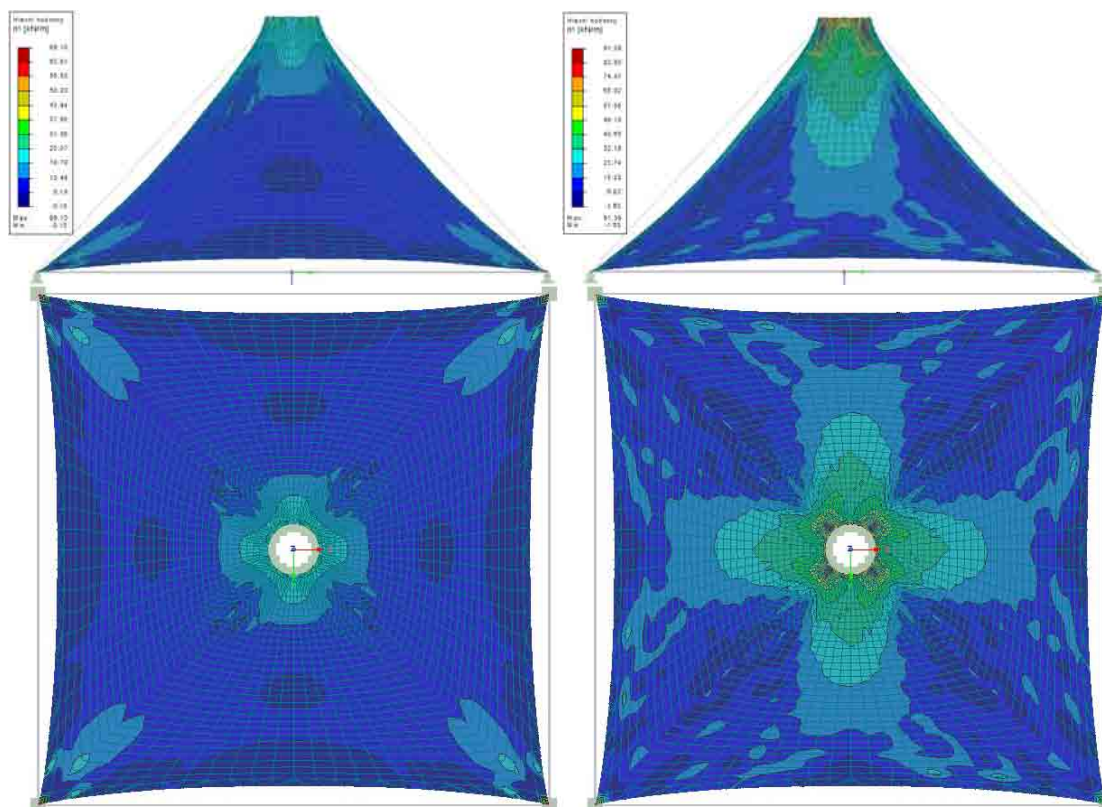
3.3.2.2.2 Poměrné přetvoření s lineární změnou ve směru osy z

3.3.2.2.2.1 Tvar 1

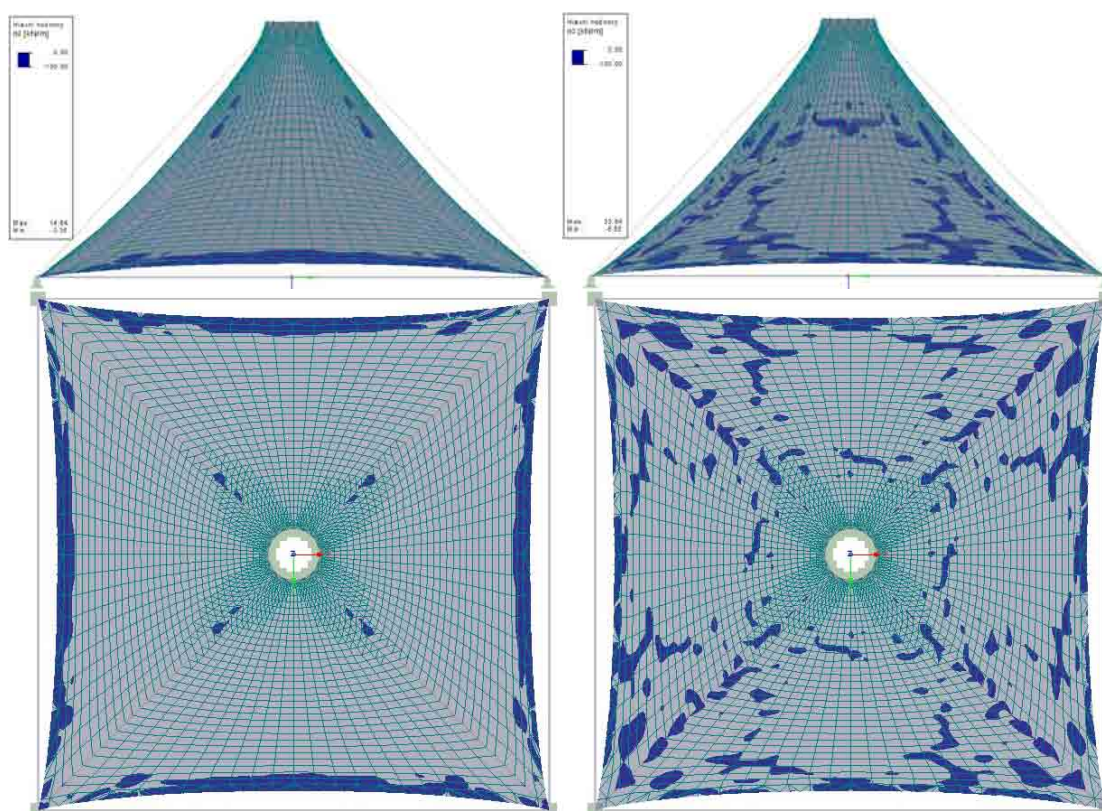
Na prvním tvaru vznikly deformace o velikosti max. 16.6 mm při stálém zatížení a při kombinaci se sněhem o velikosti 73.3 mm. Hlavní napětí n_1 vzrostlo ze 69.10 kN/m pro stálé zatížení na 91.39 kN/m v kombinaci se sněhem. V tomto případě bohužel opět nabývá hlavní napětí n_1 záporných hodnot a to -0.10 kN/m pro stálé zatížení a -1.63 kN/m pro kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -3.35 kN/m pro stálé zatížení na -5.52 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 84 Deformace při stálém zatížení a kombinaci stálého se sněhem



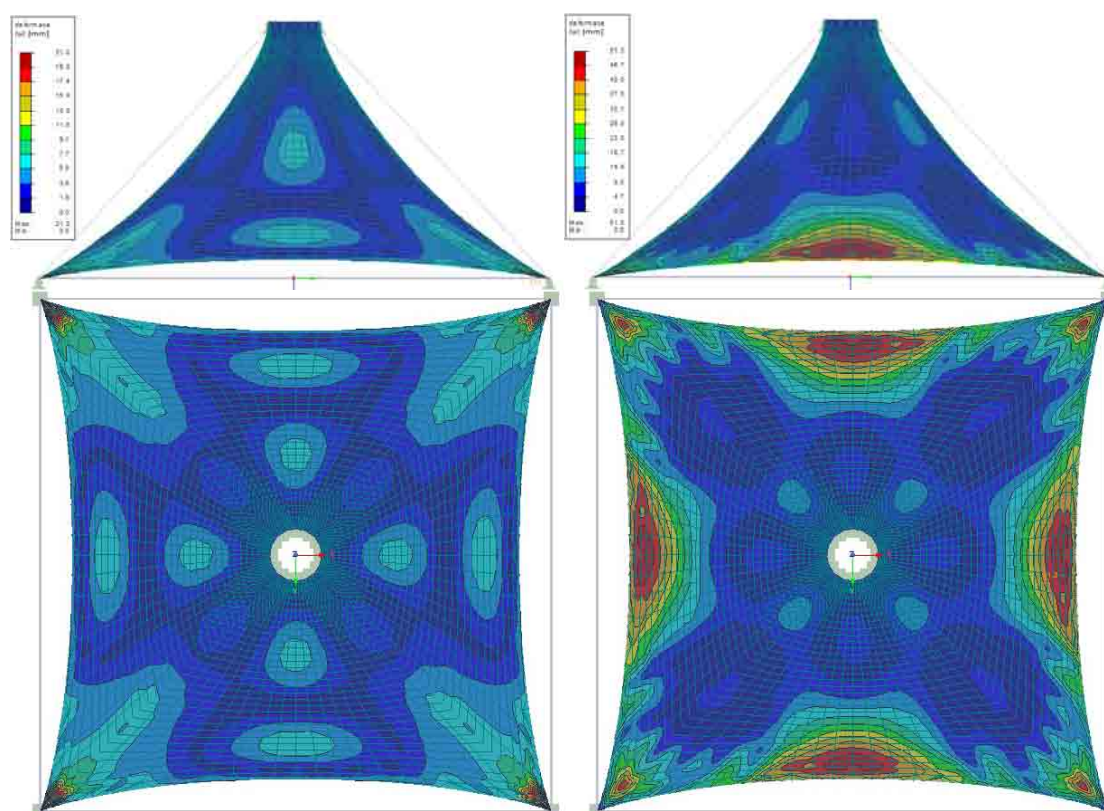
Obr. 85 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



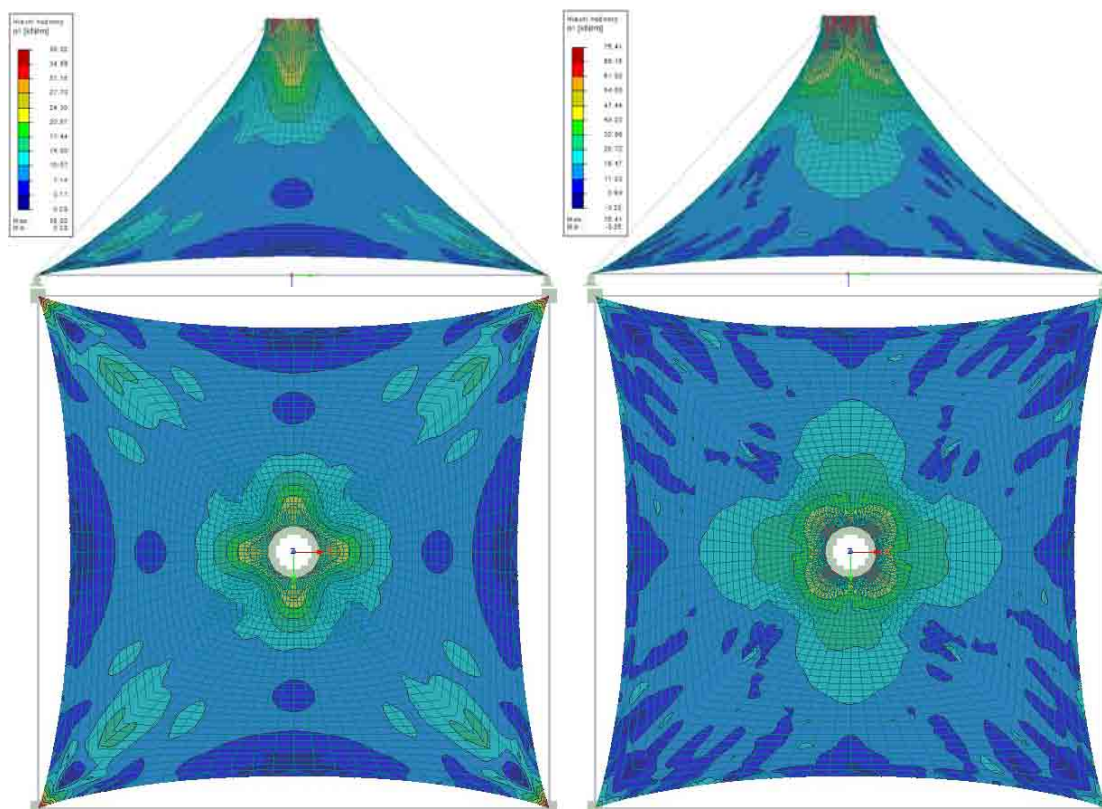
Obr. 86 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.2.2 Tvar 2

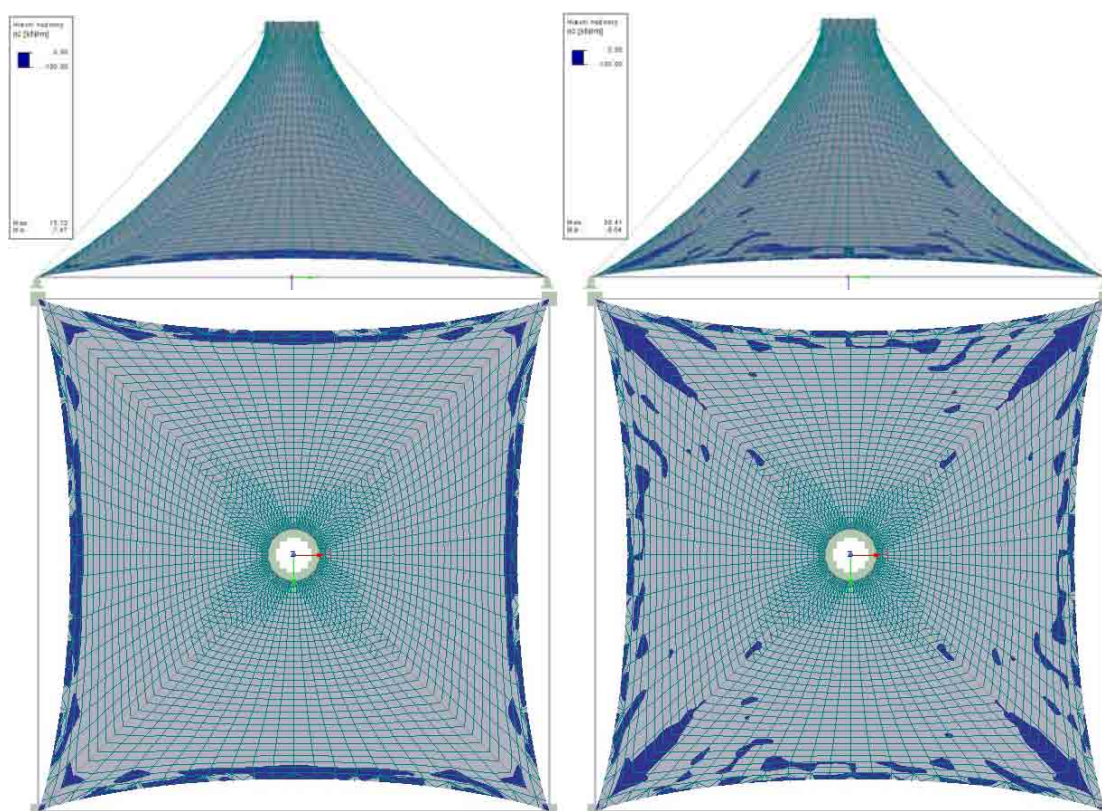
Na druhém tvaru vznikly deformace o velikosti max. 21.2 mm při stálém zatížení a při kombinaci se sněhem o velikosti 51.3 mm. Hlavní napětí n_1 vzrostlo ze 38.02 kN/m pro stálé zatížení na 76.41 kN/m v kombinaci se sněhem. Hlavní napětí n_1 pro kombinaci se sněhem dosahuje záporných hodnot a to -3.25 kN/m. Nejnižší hodnoty hlavního napětí n_2 klesly z -7.47 kN/m pro stálé zatížení na -9.04 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 87 Deformace při stálém zatížení a kombinaci stálého se sněhem



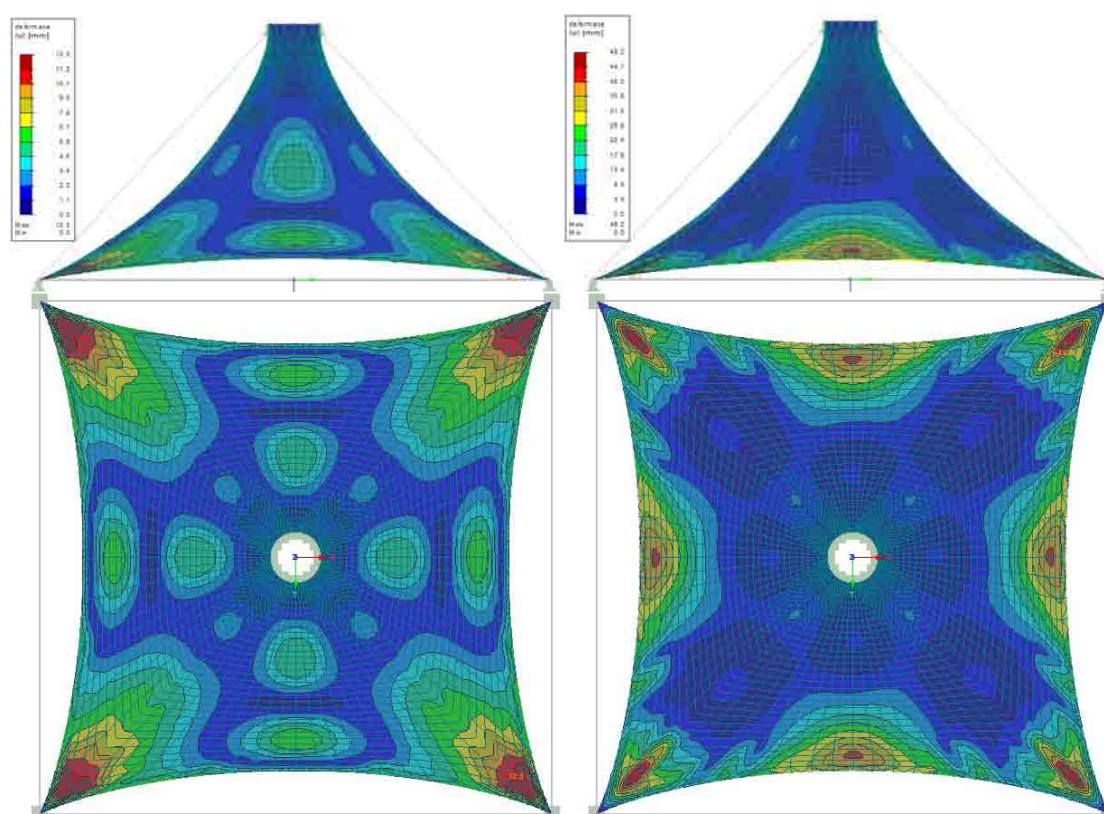
Obr. 88 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



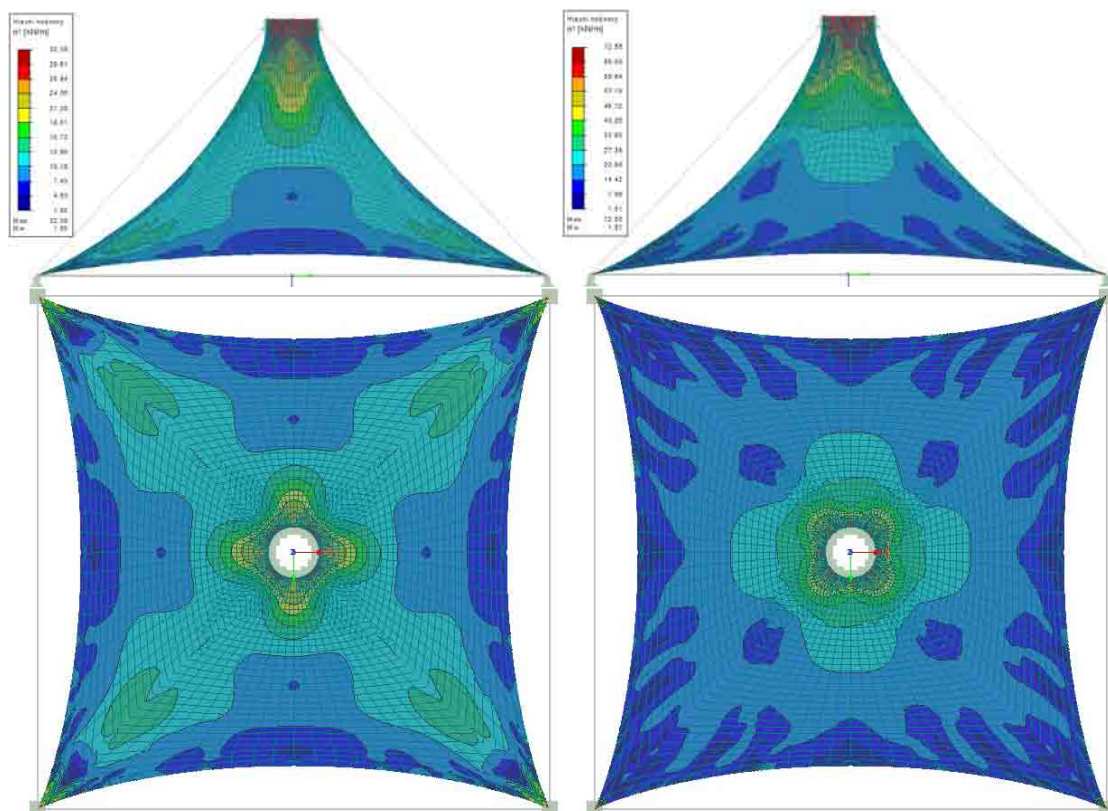
Obr. 89 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.2.2.3 Tvar 3

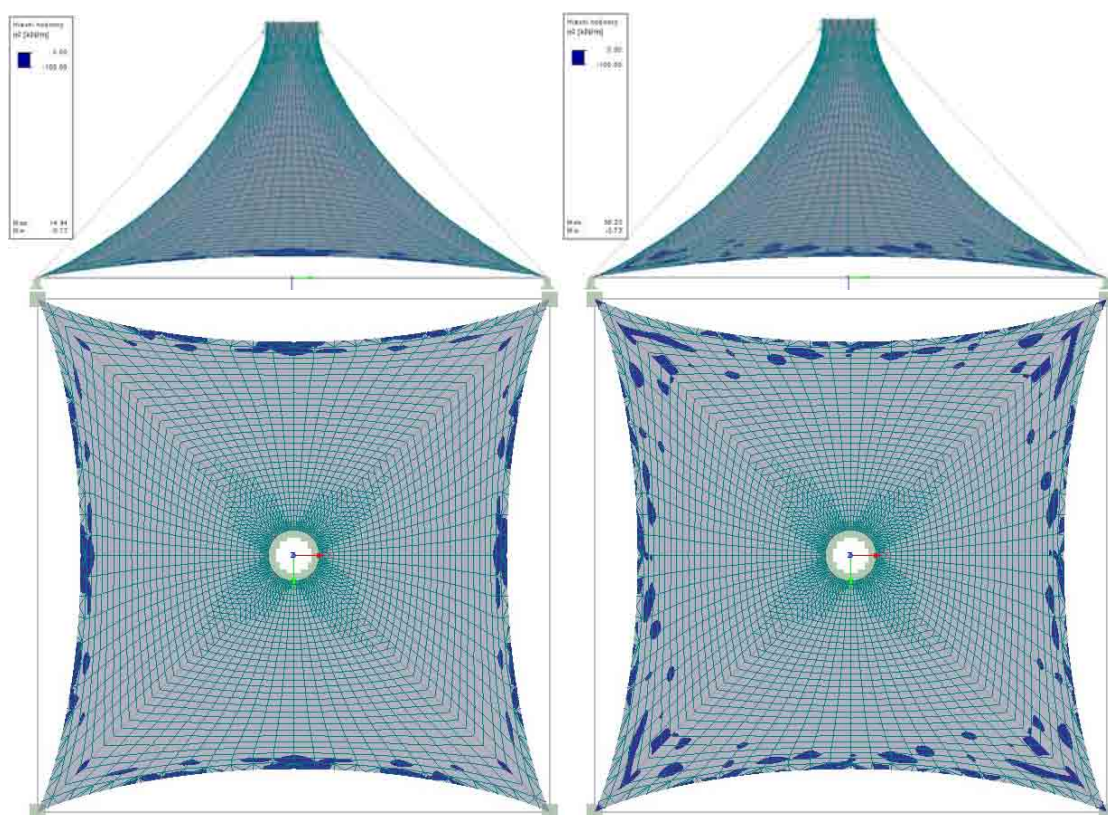
Na třetím tvaru vznikly deformace o velikosti max. 12.3 mm při stálém zatížení a při kombinaci se sněhem o velikosti 49.2 mm. Hlavní napětí n_1 vzrostlo ze 32.39 kN/m pro stálé zatížení na 72.55 kN/m v kombinaci se sněhem. Nejnižší hodnoty hlavního napětí n_2 klesly z -0.77 kN/m pro stálé zatížení na -3.72 kN/m pro kombinaci se sněhem. Rozložení deformací a napětí na ploše posuzovaného tvaru je patrné z následujících obrázků.



Obr. 90 Deformace při stálém zatížení a kombinaci stálého se sněhem



Obr. 91 Hlavní napětí n_1 při stálém zatížení a kombinaci stálého se sněhem



Obr. 92 Hlavní napětí n_2 při stálém zatížení a kombinaci stálého se sněhem

3.3.3 Shrnutí

Jak je zjevné z obr. 90-92, nejvhodnějším návrhem je tvar 3. Tento tvar měl dokonce menší deformace, než tvar vytvořený programem FormFinder. Hlavní napětí sice dosahovalo vyšších hodnot, stále však by bylo možné použít stejnou textilií jako pro výše zmíněný tvar (Valmex FR 900). Hlavní napětí n_2 také nabývalo menších hodnot.

V průběhu mé práce při hledání tvaru v programu RFEM jsem vysledoval následující možnosti vylepšení programu RFEM:

- Chybí převod geometrie membrány do výkresů konstrukce (pro další možnost zpracování, např. CAD programy).
- Pruty jsou soudržné s přilehlými plochami. Proto do nich při předpětí vnáší nežádoucí tlakové namáhání. Vhodné by bylo vytvořit kontaktní prvky 1D, které by se daly vložit mezi prut a plochu, aby byl umožněn posun prutu podél plochy, aniž by to ovlivnilo její napjatost.
- Při tvorbě Nurbs ploch se vybírají okrajové NURBS křivky a automaticky se následně vytvoří síť řídicích bodů – není možné jednoduše vybrat body a přiřadit. Je třeba každý zvlášť ručně přesunout na požadované místo. Druhou věcí je odlišná geometrie NURBS plochy a původní síť, která vyplývá z definice NURBS plochy (řídicí body neleží na ploše). Možnost automatické adjustace plochy tak, aby procházela danými body by usnadnila práci při modelování.
- U ploch s velkým rozdílem délek protilehlých linií nevzniká strukturovaná síť. Řešením by byla možnost lineárního zmenšování velikosti KP směrem ke kratší linii.
- Pro ještě lepší možnosti při hledání vhodných tvarů membránových konstrukcí by byla možnost nastavení poměrného přetvoření s různou hodnotou ve dvou směrech, případně možnost nastavení požadovaného napětí (opět s různou hodnotou ve dvou směrech).
- Pro zkvalitnění grafických výstupů bych navrhoval možnost nastavení kvality a přímý export obrázků do základních grafických formátů (možnost volit rozlišení – ne jen maximální) a možnost vypnout zobrazení sítě KP.

4. ZÁVĚR

Ve své práci jsem měl za úkol provést studii vhodných tvarů membránových konstrukcí na základě dnešních metodik návrhu konstrukcí za použití dostupných programů FormFinder a Rhinoceros. Při návrhu jednotlivých tvarů jsem se snažil maximálně využít možností jednotlivých programů k nalezení co nejvhodnějšího tvaru. Tvary vytvořené pomocí těchto programů jsem následně analyzoval v programu RFEM.

Vzhledem k výsledkům provedených analýz si odvažuji tvrdit, že tyto programy jsou schopné poskytnout vhodnou geometrii pro návrh membránových konstrukcí. Obzvláště pak program FormFinder, jehož tvary dosahovaly při analýze vynikajících výsledků.

Užitečným se mi také jeví k návrhu tvaru membránových konstrukcí program RFEM, který sice není primárně pro tyto účely určen, avšak výsledky jsou srovnatelné s výstupy programu FormFinder. Pokud by byl k programu RFEM vytvořen další modul, určený přímo k hledání vhodného tvaru, ať už pomocí metody dynamické relaxace, nebo metody Updated Reference Strategy, stal by se univerzálním nástrojem pro návrh těchto specifických konstrukcí.

Další částí procesu návrhu membránových konstrukcí je také převedení 3D tvaru membrány do dílců, které jsou následně navzájem spojeny. Tato část návrhu je nazývána Patterning a je nedílnou součástí návrhu membránových konstrukcí. Proto bych ji doporučoval zahrnout do vývoje příslušného modulu pro návrh tvarů membránových konstrukcí. Tato problematika však nebyla součástí mé diplomové práce.

5. LITERATURA

- [1] J. S. Brew and W. J. Lewis: Computational form-finding of tension membrane structures—Non-finite element approaches: Part 1. Use of cubic splines in finding minimal surface membranes, International Journal for Numerical Methods in Engineering, pp. 653-655, 2003
- [2] Rhinoceros Help
- [3] Němec et al. Finite Element Analysis of Structures
- [4] Procházková J., Vitásek V.: Základy počítačové grafiky, 2007, 56 s
mathonline.fme.vutbr.cz/pg/flash/TeorieGrafika/celaGrafika.pdf

6. SEZNAM PŘÍLOH

Příloha č.1	MeshRelaxation.rvb..
Příloha č.2	NurbsRelaxation.rvb
Příloha č.3	Katalog Pfeifer (str. 28, 33 a 38)
Příloha č.4	Katalog Mehler (Valmex FR 700, 900, 1000 a 1400)

Příloha č. 1 – MeshRelaxation.rvb

'Version december 3rd 2004

'Routines developed and copyrighted by Gelfling '04 aka. David Rutten
'Parts of these routines were developed while I was a intern at Prat-SA, Toulouse, France
'Surface relaxation using FDM (force density method) and VR(vibration reduction)
'(This method was written specifically for Rhinoceros 3.x and it's scripting engine.
'However only the interface section of the source code *requires* RhinoScript methods...)

'Internal mesh definitions differ from those required by the Rhinoceros 3.x Scripting engine.
'Instead of faces these routines use edges. Translation from one definition to the other will have to be performed prior to and after relaxation. Since only the coordinates of mesh vertices (or nodes) will be altered, this should be fairly straightforward.

'Furthermore in these routines vector objects play an important role. Vectors are arrays of 2 3D point definitions:
' Dim vecExample(1)
' vecExample(0) = Array(0,0,0) 'Starting point of vector
' vecExample(1) = Array(1,0,1) 'Ending point of vector
'Vector variables may be bigger than these dimensions but not smaller. They always need at least an x, y and z component.

Option Explicit

'This function performs a number of relaxation steps on the nodes of a mesh-definition.
'arrNodes is an array of 3D-points identifying the node coordinates in euclidian space
'arrConnections is an array of arrays containing information about connections between nodes
' Every index in the array is another array containing 2 numeric values:
' 0 = the index of the node where the connection links to
' 1 = the tension of the connection.
' Note that arrNodes and arrConnections must have the same size (every index describes a single node)
' Also note that if connections link to non-existent nodes an overflow error will occur.
' Then note that every node needs at least one connection, although logically 2 is the absolute minimum.
' Nodes can be connected to themselves, but this makes no sense so try to avoid it.
' arrConstraints is an array that describes the constraints a node can have in space.
' Every entry describes a single node. It can have the following values:
' Non-UUID string = the point is free to move in space if there is no object that has an ID which is identical to the supplied string. Using zero length strings to make sure nodes are free.
' UUID-String = the point is locked to an object (can be curve, surface or polysurface) by ClosestPoint (ClosestPoint routines are not included in this routine.
' Instead native Rhino 3.x script methods are called. If you intend to run this script on alternate platform you'll have to translate these calls into native methods. Also if other platforms use different object identifiers (instead of strings), you should make further alterations).
' Damping = a real number controlling the damping of the relaxation routine. Numbers higher than one will accelerate the relaxation, likely causing instable results. This however can be combatted by limiting the maximum distance nodes are allowed to travel. Numbers between zero and one will decelerate the relaxation thus preventing bouncing solutions, but more steps will be required to reach a stable solution within tolerance. Negative numbers should be avoided.
' Limit = a real number controlling the distance every node is allowed to move during each iteration.
' Summary will be filled with an array of data regarding the details of the entire relaxation process
' it will be filled with the following values:
' 0 = The sum-total of distances of all node translations
' 1 = The biggest resultant vector in the set
' The return value of the function is an array identical to arrNodes but with different coordinates
Private Function RelaxMeshNodes(ByRef arrNodes, ByRef arrConnections, ByRef arrConstraints, ByVal Damping, ByVal Limit, ByRef Summary)
-
 Dim i, j
 Dim arrV(), resV
 Dim newNodes
 Dim ptCP

 Dim maxResVec, curResVec
 Dim sumNodeTrans

 maxResVec = 0.0
 sumNodeTrans = 0.0

```

newNodes = arrNodes
For i = 0 To UBound(newNodes)
    If arrConstraints(i) <> "FIXED" Then
        Erase arrV
        ReDim arrV(UBound(arrConnections(i)))
        For j = 0 To UBound(arrV)
            arrV(j) = MultiplyVector(Array(newNodes(i),
newNodes(arrConnections(i)(j)(0))), _
arrConnections(i)(j)(1))
        Next
        resV = ResultantVector(arrV)
        resV = MultiplyVector(resV, Damping)
        resV = LimitVector(resV, Limit)

        If Rhi no. IsObject(arrConstraints(i)) Then
            If Rhi no. IsCurve(arrConstraints(i)) Then
                ptCP = Rhi no. EvaluateCurve(arrConstraints(i),
Rhi no. CurveClosestPoint(arrConstraints(i), resV(1)))
            Else If Rhi no. IsBREP(arrConstraints(i)) Then
                ptCP = Rhi no. BrepClosestPoint(arrConstraints(i),
resV(1))(0)
            End If

            If IsArray(ptCP) Then
                newNodes(i) = ptCP
                curResVec = PointPointDistance(newNodes(i),
arrNodes(i))
                If curResVec > maxResVec Then maxResVec =
curResVec
                sumNodeTrans = sumNodeTrans + curResVec
            Else
                newNodes(i) = resV(1)
                curResVec = PointPointDistance(newNodes(i),
arrNodes(i))
                If curResVec > maxResVec Then maxResVec =
curResVec
                sumNodeTrans = sumNodeTrans + curResVec
            End If
        Else
            newNodes(i) = resV(1)
            curResVec = PointPointDistance(newNodes(i),
arrNodes(i))
            If curResVec > maxResVec Then maxResVec = curResVec
            sumNodeTrans = sumNodeTrans + curResVec
        End If
    End If
Next

Summary = Array(sumNodeTrans, maxResVec)
RelaxMeshNodes = newNodes
End Function

Private Function PointPointDistance(ByRef arrPoint1, ByRef arrPoint2)
    PointPointDistance = (arrPoint1(0)-arrPoint2(0)) * (arrPoint1(0)-arrPoint2(0)) + _
(arrPoint1(1)-arrPoint2(1)) *
(arrPoint1(1)-arrPoint2(1)) + _
(arrPoint1(2)-arrPoint2(2)) *
(arrPoint1(2)-arrPoint2(2))
    PointPointDistance = Sqr(PointPointDistance)
End Function

' This function calculates the resultant vector of an array of vectors.
' The starting point of the first vector in the array will be used as grip
' for the resultant vector, if an invalid vector array is passed an error will occur
Private Function ResultantVector(ByRef arrVectors)
    Dim i
    Dim di vX, di vY, di vZ
    Dim resX, resY, resZ
    Dim ptT(2), ptH(2)

    resX = 0.0
    resY = 0.0
    resZ = 0.0
    For i = 0 To UBound(arrVectors)
        di vX = arrVectors(i)(1)(0) - arrVectors(i)(0)(0)
        di vY = arrVectors(i)(1)(1) - arrVectors(i)(0)(1)
        di vZ = arrVectors(i)(1)(2) - arrVectors(i)(0)(2)
        resX = resX + di vX
        resY = resY + di vY
        resZ = resZ + di vZ
    Next
    ptT(0) = arrVectors(0)(0)(0)
    ptT(1) = arrVectors(0)(0)(1)

```



```

    ptT(2) = arrVectors(0)(0)(2)
    ptH(0) = ptT(0) + resX
    ptH(1) = ptT(1) + resY
    ptH(2) = ptT(2) + resZ

    ResultantVector = Array(ptT, ptH)
End Function

' This function will limit a vector length to a certain value
' vectors smaller than the specified length will remain untouched.
' Note that it is unhealthy to use negative numbers or zero for dblMaxLength
Private Function LimitVector(ByVal vecIn, ByVal dblLimitation)
    Dim vecOut(1)
    Dim l

    vecOut(0) = vecIn(0)
    vecOut(1) = vecIn(1)
    l = PointPointDistance(vecIn(0), vecIn(1))

    If l > dblLimitation Then
        vecOut(1)(0) = vecOut(0)(0) + ((vecOut(1)(0)-vecOut(0)(0)) / l) *
dblLimitation
        vecOut(1)(1) = vecOut(0)(1) + ((vecOut(1)(1)-vecOut(0)(1)) / l) *
dblLimitation
        vecOut(1)(2) = vecOut(0)(2) + ((vecOut(1)(2)-vecOut(0)(2)) / l) *
dblLimitation
    End If
    LimitVector = vecOut
End Function

' This function will multiply a vector length by a specified factor.
Private Function MultiplyVector(ByVal vecIn, ByVal dblFactor)
    Dim vecOut(1)
    vecOut(0) = vecIn(0)
    vecOut(1) = vecIn(1)
    vecOut(1)(0) = vecOut(0)(0) + (vecOut(1)(0) - vecOut(0)(0)) * dblFactor
    vecOut(1)(1) = vecOut(0)(1) + (vecOut(1)(1) - vecOut(0)(1)) * dblFactor
    vecOut(1)(2) = vecOut(0)(2) + (vecOut(1)(2) - vecOut(0)(2)) * dblFactor
    MultiplyVector = vecOut
End Function

' -----
' -----
' -----
' -----
' -----Thus ends the mesh relaxation source
code...-----
' -----Below you will find an subroutine which implements the above
routine-----
' -----and also adds interface code. (This subroutine requires
Rhino.ceros 3.x)-----
' -----
' -----
' -----
' -----

' This function translates a Rhino Mesh Surface definition into an FDM Mesh definition
' If the function succeeds True will be returned, if the function fails Null will be
returned.
Private Function MESH2FDM(ByVal idMesh)
    MESH2FDM = Null

    Dim i, j, N, P1, P2
    Dim arrCString(), arrC(), arrL()
    Dim SomeEdges, AllEdges()
    Dim vReturn, vSplit, iSplit, arrTemp()

    ' These are properties of Rhino meshes, no need to calculate them ourselves
    arrNodes = Rhino.MeshVertices(idMesh)
    arrFaces = Rhino.MeshFaceVertices(idMesh)

    ' Load data from object (Only if the mesh was relaxed before)
    If IsMeshCompliant(idMesh) Then
        ' If MsgBox("There exists relaxation data from a previous run. Do you want
to load this data?", vbYesNo Or vbQuestion, "Existing data") = vbYes Then
            ReDim arrC(UBound(arrNodes))
            ReDim arrL(UBound(arrNodes))
            Rhino.Prompt "Creating dummy arrays..."
            For i = 0 To UBound(arrNodes)
                arrC(i) = Array(0, 10)
                arrL(i) = "FREE"
            Next i
        End If
    End If
End Function
```

```

        Next
        arrConnections = arrC
        arrConstraints = arrL
        Rhino.Prompt "Loading settings from mesh-object..."
        For i = 0 To UBound(arrNodes)
            LoadNode idMesh, i
        Next
        MESH2FDM = True
        Exit Function
    End If

    'Resize the Link and Connection arrays to match the Vertex array
    ReDim arrL(UBound(arrNodes))
    ReDim arrCString(UBound(arrNodes))

    'Load vertex link properties from the object USERDATA. If no data could be found
    the node defaults to FREE
    For i = 0 To UBound(arrL)
        vReturn = Rhino.GetObjectData(idMesh, "Gefling_Relaxation", "VertexLink" &
CStr(i))
        If IsNull(vReturn) Then
            arrL(i) = "FREE"
        Else
            arrL(i) = vReturn
        End If
    Next

    Rhino.Prompt "Creating all meshvertex connections..."
    N = 0
    For i = 0 To UBound(arrFaces)
        'Add edge 1 of face i
        ReDim Preserve AllEdges(N): AllEdges(N) =
CStr(Min(arrFaces(i)(0), arrFaces(i)(1))) & ";" & CStr(Max(arrFaces(i)(0), arrFaces(i)(1)))
        N = N+1
        'Add edge 2 of face i
        ReDim Preserve AllEdges(N): AllEdges(N) =
CStr(Min(arrFaces(i)(1), arrFaces(i)(2))) & ";" & CStr(Max(arrFaces(i)(1), arrFaces(i)(2)))
        N = N+1
        'If the face is a quad then add edge 3 of face i
        If arrFaces(i)(2) <> arrFaces(i)(3) Then
            ReDim Preserve AllEdges(N): AllEdges(N) =
CStr(Min(arrFaces(i)(2), arrFaces(i)(3))) & ";" & CStr(Max(arrFaces(i)(2), arrFaces(i)(3)))
            N = N+1
        End If
        'Add the closing edge of face i
        ReDim Preserve AllEdges(N): AllEdges(N) = CStr(Min(arrFaces(i)(0), arrFaces(i)(3)))
        & ";" & CStr(Max(arrFaces(i)(0), arrFaces(i)(3)))
        N = N+1
    Next

    Rhino.Prompt "Removing duplicate connections..."
    'Since all out edges have the lowest index first we can use a native function to
    remove all duplicates.
    SomeEdges = Rhino.CullDuplicateStrings(AllEdges)

    Rhino.Prompt "Creating connection lists..."
    For i = 0 To UBound(SomeEdges)
        'Try to find the tension factor of the connection, if the data is not
        available the connection defaults to 10.0
        vReturn = Rhino.GetObjectData(idMesh, "Gefling_Relaxation",
"ConnectionStrength(" & SomeEdges(i) & ")")
        If IsNull(vReturn) Then vReturn = 10.0
        'Split the string into the individual parent indices
        vSplit = Split(SomeEdges(i), ";")
        P1 = vSplit(0)
        P2 = vSplit(1)
        'Add the connection to both parents (bidirectional, this saves time later)
        arrCString(P1) = arrCString(P1) & P2 & ";" & vReturn & Space(2)
        arrCString(P2) = arrCString(P2) & P1 & ";" & vReturn & Space(2)
    Next

    Rhino.Prompt "Parsing connection lists..."
    'Resize the connection array to match the size of the stringconnection array
    ReDim arrC(UBound(arrCString))
    For i = 0 To UBound(arrCString)
        'Remove the last two spaces (actually in case of an isolated node the
        script will crash, make sure we only feed it proper meshes)
        arrCString(i) = Trim(arrCString(i))
        'Break all the connections apart
        vSplit = Split(arrCString(i), Space(2))

        ReDim Preserve arrTemp(UBound(vSplit))
        For j = 0 To UBound(vSplit)

```

```

' Break the individual connections into Parent-index and Tension
value
    iSplit = Split(vSplit(j), ";")
    'Add these values to the temporal array
    arrTemp(j) = Array(CInt(iSplit(0)), CDBl(iSplit(1)))
Next
    'Add the temporal array to the connection array @ vertex i
    arrC(i) = arrTemp
Next

Rhino.Prompt "Done building FDM-mesh..."
arrConnections = arrC
arrConstraints = arrL
MESH2FDM = True
End Function

Private Function Min(Value1, Value2)
    Min = Value1
    If Value2 < Value1 Then Min = Value2
End Function

Private Function Max(Value1, Value2)
    Max = Value1
    If Value2 > Value1 Then Max = Value2
End Function

' This function removes all objects from the document that have been created by functions in
this script.
' Basically it removes all objects that are named "GelflingRelaxationPreviewObjects"
Private Sub ClearPreview()
    Dim allPreviewObjects
    allPreviewObjects = Rhino.ObjectsByName("GelflingRelaxationPreviewObjects", False)
    If IsNull(allPreviewObjects) Then Exit Sub
    Rhino.DeleteObjects allPreviewObjects
End Sub

' This function asks the user to select any amount of nodes from the mesh.
' The return value is NULL if the user did not select any nodes or on error,
' The return value is an array of node indices if successful
Private Function GetMeshNodes(idMesh, strMessage, FilterTypology)
    GetMeshNodes = Null
    Dim arrNaked
    Dim selGrips, i
    Dim arrSelected(), S

    Rhino.EnableRedraw False
    Rhino.EnableObjectGrips idMesh
    arrNaked = Rhino.MeshNakedEdgePoints(idMesh)
    Select Case UCase(FilterTypology)
        Case "EXTERIOR"
            For i = 0 To UBound(arrNaked)
                If Not arrNaked(i) Then
                    Rhino.SelectObjectGrip idMesh, i
                End If
            Next
            Rhino.Command "-_HidePt", vbFalse
        Case "INTERIOR"
            For i = 0 To UBound(arrNaked)
                If arrNaked(i) Then
                    Rhino.SelectObjectGrip idMesh, i
                End If
            Next
            Rhino.Command "-_HidePt", vbFalse
        Case Else
            ' Show all grips
    End Select
    Rhino.EnableRedraw True

    selGrips = Rhino.GetObjectGrips(strMessage, False, True)
    Rhino.EnableRedraw False
    Rhino.EnableObjectGrips idMesh, False
    Rhino.Command "-_ShowPt", vbFalse
    Rhino.EnableRedraw True

    If IsNull(selGrips) Then Exit Function

    S = 0
    For i = 0 To UBound(selGrips)
        If selGrips(i)(0) = idMesh Then
            ReDim Preserve arrSelected(S)
            arrSelected(S) = selGrips(i)(1)
            S = S + 1
        End If
    Next
    GetMeshNodes = arrSelected

```

```

End Function

Private Function IsNodeConnectedTo(Byval MainIndex, ByVal SearchIndex)
    IsNodeConnected = False
    Dim i
    For i = 0 To UBound(arrConnections(MainIndex))
        If arrConnections(MainIndex)(i)(0) = SearchIndex Then
            IsNodeConnected = True
            Exit Function
        End If
    Next
End Function

Private Sub SetNode(Byval idMesh, ByVal NodeIndex)
    Dim vInfo, i
    vInfo = arrConstraints(NodeIndex)
    For i = 0 To UBound(arrConnections(NodeIndex))
        vInfo = vInfo & ":" & arrConnections(NodeIndex)(i)(0) & "_" &
arrConnections(NodeIndex)(i)(1)
    Next
    ' Rhino.Print vInfo
    Rhino.SetObjectData idMesh, "ReconstructivismRelaxation", _
        "Node:" & NodeIndex, vInfo
End Sub

Private Function LoadNode(Byval idMesh, ByVal NodeIndex)
    Dim vInfo, i
    Dim aSplit, tSplit
    Dim arrT()

    vInfo = Rhino.GetObjectData(idMesh, "ReconstructivismRelaxation", "Node:" &
NodeIndex)
    If IsNull(vInfo) Then Exit Function
    aSplit = Split(vInfo, ":")
    If Not IsArray(aSplit) Then Exit Function

    ReDim arrT(UBound(aSplit)-1)
    arrConstraints(NodeIndex) = aSplit(0)
    For i = 1 To UBound(aSplit)
        tSplit = Split(aSplit(i), "_")
        arrT(i-1) = Array(CLng(tSplit(0)), CDBl(tSplit(1)))
    Next
    arrConnections(NodeIndex) = arrT
    LoadNode = True
End Function

Private Function PurgeAllInfo(Byval idMesh)
    Rhino.DeleteObjectData idMesh, "ReconstructivismRelaxation"
End Function

Private Sub StoreCurrentHashCode(Byval idMesh)
    Rhino.SetObjectData idMesh, "ReconstructivismRelaxation", "HashCode",
MeshHashCode(idMesh)
End Sub

Private Function IsMeshCompliant(Byval idMesh)
    IsMeshCompliant = False
    Dim nHash, sHash
    nHash = MeshHashCode(idMesh)
    sHash = Rhino.GetObjectData(idMesh, "ReconstructivismRelaxation", "HashCode")
    If IsNull(sHash) Then Exit Function
    If sHash <> nHash Then Exit Function
    IsMeshCompliant = True
End Function

Private Function MeshHashCode(Byval idMesh)
    Dim arrN, arrF
    arrN = Rhino.MeshVertices(idMesh)
    arrF = Rhino.MeshFaceVertices(idMesh)
    MeshHashCode = "???" & UBound(arrN) & "_" & UBound(arrF) & "???"
End Function

' This is it. The main sub...
Dim arrNodes                ' An array containing all mesh node coordinates
Dim arrFaces                ' An array containing all face definitions (not used during
relaxation)
Dim arrTexture              ' An array containing all texture coordinates for the
material bitmap
Dim arrConnections          ' An array containing connectivity information for every
Node (index0 = Other node, index1 = connection tension)
Dim arrConstraints          ' An array containing per node constraint information
(Either "FREE", "FIXED" or ValidRhinoObjectID)

Dim idMaterial              ' A long value indicating the tension texture material

```



```

Dim BasePath 'A string indicating where the texture is located
Private Sub RelaxMESH_Surface()
    'Create base-path settings
    BasePath = Rhino.GetSettings(Rhino.InstallFolder & "RelaxData.ini",
"RECONSTRUCTIVISMRELAXATION", "BASEFOLDER")
    If IsNull(BasePath) Then
        Dim fso : Set fso = CreateObject("Scripting.FileSystemObject")
        BasePath = Rhino.BrowseForFolder(Rhino.InstallFolder, "Select the
folder for this script...", "Relaxation")
    End If
    Set fso = Nothing
    Rhino.SaveSettings Rhino.InstallFolder & "RelaxData.ini",
"RECONSTRUCTIVISMRELAXATION", "BASEFOLDER", BasePath
    'Finish base path setting

    Dim strMeshID, newMeshID, PreviewMeshID
    Dim DampingFactor, Limitation, PreflightLoops, AccuracyMark, LimitationDecay
    Dim Summary, DisplayMode, ActiveView
    Dim MeshBBox, Diagonal
    Dim RunAsDeveloper, DestinationFolder, FramePrefix, FrameFormat, FrameDim(1)

    RunAsDeveloper = False

    strMeshID = Rhino.GetObject("Select a mesh for relaxation", 32, True, True)
    If IsNull(strMeshID) Then Exit Sub

    Rhino.Prompt "Loading geometry... please wait"
    Summary = MESH2FDM(strMeshID)
    If IsNull(Summary) Then
        MsgBox "Error in building FDM mesh.", vbOkOnly Or vbCritical, "Relaxation
error"
    End If
    Exit Sub

    Dim arrOptions, strResult
    Dim intObject, i, j, N
    Dim idParent, Node
    Dim idObjects

    Do
        If RunAsDeveloper Then
            arrOptions = Array("Tension_Factors", "Geometry_Links", "Summary",
"Relax", "Relax_advanced", "Quit")
        Else
            arrOptions = Array("Tension_Factors", "Geometry_Links", "Relax",
"Quit")
        End If
        strResult = Rhino.GetString("Relaxation settings", "Relax", arrOptions)
        If IsNull(strResult) Then Exit Sub
        Select Case Left(UCase(strResult), 1)
            Case "M"
                If UCase(strResult) = "MELLON" Then RunAsDeveloper = True
            Case "T"
                strResult = Rhino.GetString("Select a filter for spring
selection...", "All", "Interior", "Exterior"))
                If Not IsNull(strResult) Then
                    idObjects = GetMeshNodes(strMeshID, "Select nodes to
retension...", strResult)
                    If IsArray(idObjects) Then
                        strResult = Rhino.GetReal("Specify a new tension
factor", 10.0, 0.01, 10000.0)
                        If Not IsNull(strResult) Then
                            For Each Node in idObjects
                                For i = 0 To
                                    UBound(arrConnections(Node))
                                        For j = 0 To
                                            UBound(idObjects(j))
                                                If idObjects(j) =
arrConnections(Node)(i)(0) Then
                                                    arrConnections(Node)(i)(1) = strResult
                                                End If
                                            Next
                                        Next
                                    Next
                                End If
                            End If
                        End If
                    End If
                Case "G"
                    strResult = Rhino.GetString("Select a filter for node
selection...", "All", "Interior", "Exterior"))
                    If Not IsNull(strResult) Then
                        idObjects = GetMeshNodes(strMeshID, "Select nodes to

```

```

(un)constrain...", strResult)
    If IsArray(i dObjects) Then
        strResult = Rhino.GetString("Pick a node restraint
type", "Fixed", Array("Fixed", "Free", "Linked"))
        If Not IsNull(strResult) Then
            Select Case Left(UCase(strResult), 2)
                Case "FR"
                    For i = 0 To UBound(i dObjects)

arrConstraints(i dObjects(i)) = "FREE"

                Next
            Case "LI"
                i dParent = Rhino.GetObject("Select
a guide object", 4+8+16, False, False)
                If Not IsNull(i dParent) Then
                    For i = 0 To

UBound(i dObjects)

arrConstraints(i dObjects(i)) = i dParent

                Next
            Case Else
                For i = 0 To UBound(i dObjects)

arrConstraints(i dObjects(i)) = "FIXED"

                Next
            End Select
        End If
    End If
End If
Case "S"
    N = 0
    For i = 0 To UBound(arrConnections)
        N = N + UBound(arrConnections(i))+1
    Next
    MsgBox "MESH surface relaxation data for object; " & vbNewLine & _
        "[" & strMeshID & "]" & vbNewLine & vbNewLine & _
        "Mesh node count; " & UBound(arrNodes)+1 & vbNewLine & _
        "Mesh face count; " & UBound(arrFaces)+1 & vbNewLine & _
        "Mesh link count; " & N & vbNewLine & _
        "end of summary...", vbOkOnly, "FDM-mesh summary"
Case "R"
    MeshBBox = Rhino.BoundingBox(strMeshID)
    Diagonal = PointPointDistance(MeshBBox(0), MeshBBox(6))
    If Len(strResult) = 14 Then
        AccuracyMark = Rhino.GetReal("Specify relaxation
tolerance", Rhino.UnitsAbsoluteTolerance, Rhino.UnitsAbsoluteTolerance/100)
        DampingFactor = Rhino.GetReal("Specify a global damping
factor", 1.0, 0.001, 100.0)
        Limitation = Rhino.GetReal("Specify an initial translation
limitation", Round(Diagonal * 0.025, 2), AccuracyMark, Diagonal*10)
        LimitationDecay = Rhino.GetReal("Specify a translation
limitation decay factor per iteration", 0.95, 0.1, 0.99)
        PreflightLoops = Rhino.GetInteger("Specify the number of
preflight-iterations", 20, 1, 1000)
        If IsNull(AccuracyMark) Then AccuracyMark =
Rhino.UnitsAbsoluteTolerance
        If IsNull(DampingFactor) Then DampingFactor = 1.0
        If IsNull(Limitation) Then Limitation = Diagonal * 0.025
        If IsNull(LimitationDecay) Then LimitationDecay = 0.95
        If IsNull(PreflightLoops) Then PreflightLoops = 20
    Else
        DampingFactor = 1.0
        Limitation = Diagonal * 0.025
        PreflightLoops = 30
        AccuracyMark = Rhino.UnitsAbsoluteTolerance
        LimitationDecay = 0.95
    End If
Exit Do
Case "Q"
Exit Sub
Case Else
    Rhino.Print "Unknown command entered... nothing done."
End Select
Loop
Rhino.Prompt "Writing node-data to mesh-object... (this will take a while on large
meshes)"
For i = 0 To UBound(arrNodes)
    SetNode strMeshID, i
Next
StoreCurrentHashCode strMeshID
ActiveView = Rhino.CurrentView

```

```

' DisplayMode = Rhino.ViewDisplayStyleMode(ActiveView, 2)
' AssignTextureCoords
Rhino.UnselectAllObjects
Rhino.HideObject strMeshID
PreviousMeshID = Rhino.AddMesh(arrNodes, arrFaces)
' PreviousMeshID = Rhino.AddMesh(arrNodes, arrFaces,, arrTexture)
' ApplyMaterial PreviousMeshID

DestinationFolder = Null
If RunAsDeveloper Then
    DestinationFolder = Rhino.BrowseForFolder("Select a folder to save the
animation sequence. Press 'Cancel' to not save animation.", "Animation settings")
    If Not IsNull(DestinationFolder) Then
        FramePrefix = Rhino.StringBox("Enter a name for the sequence...",
"RelaxationFrame", "Animation settings")
        If IsNull(FramePrefix) Then
            DestinationFolder = Null
        Else
            FrameFormat = Rhino.ListBox(Array("bmp", "tga", "pcx",
"jpg", "png", "tif"), "Select a file format", "Animation settings")
            If IsNull(FrameFormat) Then
                DestinationFolder = Null
            Else
                Summary = Rhino.PropertyListBox(Array("Animation
width", "Animation height"), _
Array("1500", "1000"), "Enter frame dimensions", "Animation settings")
                If IsNull(Summary) Then
                    DestinationFolder = Null
                Else
                    FrameDim(0) = Summary(0)
                    FrameDim(1) = Summary(1)
                End If
            End If
        End If
    End If

    If Not IsNull(DestinationFolder) Then
        Rhino.Command "-_TestViewCaptureToFile " & Chr(34) &
DestinationFolder & FramePrefix & "00000." & FrameFormat & Chr(34) &
" _Width=" & FrameDim(0) & "
_Height=" & FrameDim(1) & " _DrawCPlane=No _DrawWorldAxes=No _Enter", vbFalse
    End If
End If

' Analyse current FDM mesh properties
Rhino.Prompt "Performing test relaxation run... please wait"
RelaxMeshNodes arrNodes, arrConnections, arrConstraints, DampingFactor, Limitation,
Summary
Rhino.Print "Initial cumulative node translation: " & Round(Summary(0)) &
Rhino.UnitsSystemName(False, True, True) & " Initial largest node vector: " & Round(Summary(1), 2) &
" N"

' Perform preflight relaxation
For i = 1 To PreflightLoops
    Rhino.Prompt "Performing preflight relaxation. Step " & i & " of " &
PreflightLoops & "... "
    arrNodes = RelaxMeshNodes(arrNodes, arrConnections, arrConstraints,
DampingFactor, Limitation, Summary)

    Rhino.EnableRedraw vbFalse
    Rhino.DeleteObject PreviousMeshID
    PreviousMeshID = Rhino.AddMesh(arrNodes, arrFaces)
    ' PreviousMeshID = Rhino.AddMesh(arrNodes, arrFaces,, arrTexture)
    ' ApplyMaterial PreviousMeshID
    Rhino.EnableRedraw vbTrue

    If Not IsNull(DestinationFolder) Then
        Rhino.Command "-_TestViewCaptureToFile " & Chr(34) &
DestinationFolder & FramePrefix & Right("00000" & i, 5) & "." & FrameFormat & Chr(34) &
" _Width=" & FrameDim(0) & "
_Height=" & FrameDim(1) & " _DrawCPlane=No _DrawWorldAxes=No _Enter", vbFalse
    End If
Next

' Perform damped relaxation
Do
    arrNodes = RelaxMeshNodes(arrNodes, arrConnections, arrConstraints,
DampingFactor, Limitation, Summary)
    Rhino.Prompt "Iteration " & i & _
& _
& " Damping: " & Round(DampingFactor*100, 2) & "%"
& _
& " Limitation: " & Round(Limitation, 2) &
Rhino.UnitsSystemName(False, True, True)

```

```

        Limitation = Limitation * LimitationDecay
        Rhino.EnableRedraw vbFalse
        Rhino.DeleteObject PreviewMeshID
        PreviewMeshID = Rhino.AddMesh(arrNodes, arrFaces)
        ' PreviewMeshID = Rhino.AddMesh(arrNodes, arrFaces,, arrTexture)
        ' ApplyMaterial PreviewMeshID
        Rhino.EnableRedraw vbTrue

        If Not IsNull(DestinationFolder) Then
            Rhino.Command "-_TestViewCaptureToFile " & Chr(34) &
            DestinationFolder & FramePrefix & Right("00000" & i, 5) & ". " & FrameFormat & Chr(34) & _
            " _Width=" & FrameDim(0) & " _
            _Height=" & FrameDim(1) & " _DrawCPlane=No _DrawWorldAxes=No _Enter", vbFalse
        End If

        If Limitation < AccuracyMark Then Exit Do
        i = i + 1
    Loop

    ' Rhino.ViewDisplayMode ActiveView, DisplayMode
    Rhino.EnableRedraw False
    newMeshID = Rhino.AddMesh(arrNodes, arrFaces)

    Rhino.Prompt "Writing node-data to mesh-object... please wait"
    For i = 0 To UBound(arrNodes)
        SetNode newMeshID, i
    Next
    StoreCurrentHashCode newMeshID
    Rhino.ShowObject strMeshID
    Rhino.DeleteObject PreviewMeshID
    Rhino.DeleteObject strMeshID
    Rhino.SelectObject newMeshID
    Rhino.EnableRedraw True
End Sub
RelaxMESH_Surface

' Feedback
functions-----
' Private Function ApplyMaterial (ByVal idMesh)
'     idMaterial = Rhino.AddMaterialToObject(idMesh)
'     Rhino.MaterialTexture idMaterial, BasePath & "TensionGradient.bmp"
'     ApplyMaterial = True
' End Function

' Private Function AssignTextureCoords()
'     Dim i, T
'     Dim arrT()
'     ReDim arrT(UBound(arrNodes))
'     For i = 0 To UBound(arrNodes)
'         T = GetNodeMaxTension(i)
'         If T < 0 Then T = 0
'         If T > 1000 Then T = 1000
'         T = T/1000
'         arrT(i) = Array(T, 0.5)
'     Next
'     arrTexture = arrT
'     AssignTextureCoords = True
' End Function

' Private Function GetNodeMaxTension(Byval nIndex)
'     Dim i, maxT
'     maxT = 0
'     For i = 0 To UBound(arrConnections(nIndex))
'         If arrConnections(nIndex)(i)(1) > maxT Then maxT =
arrConnections(nIndex)(i)(1)
'     Next
'     GetNodeMaxTension = maxT
' End Function

```


Příloha č. 2– NurbsRelaxation.rvb

'Version december 3rd 2004

'Routines developed and copyrighted by Gelfling '04 aka. David Rutten
'Parts of these routines were developed while I was a intern at Prat-SA, Toulouse, France
'Surface relaxation using FDM (force density method) and VR(vibration reduction)
'(This method was written specifically for Rhinoceros 3.x and it's scripting engine.
'However only the interface section of the source code *requires* RhinoScript methods...)

'Internal mesh definitions differ from those required by the Rhinoceros 3.x Scripting engine.
'Instead of faces these routines use edges. Translation from one definition to the other will have to be performed prior to and after relaxation. Since only the coordinates of mesh vertices (or nodes) will be altered, this should be fairly straightforward.

'Furthermore in these routines vector objects play an important role. Vectors are arrays of 2 3D point definitions:
' Dim vecExample(1)
' vecExample(0) = Array(0,0,0) 'Starting point of vector
' vecExample(1) = Array(1,0,1) 'Ending point of vector
'Vector variables may be bigger than these dimensions but not smaller. They always need at least an x, y and z component.

Option Explicit

'This function performs a number of relaxation steps on the nodes of a mesh-definition.
'arrNodes is an array of 3D-points identifying the node coordinates in euclidian space
'arrConnections is an array of arrays containing information about connections between nodes
' Every index in the array is another array containing 2 numeric values:
' 0 = the index of the node where the connection links to
' 1 = the tension of the connection.
' Note that arrNodes and arrConnections must have the same size (every index describes a single node)
' Also note that if connections link to non-existent nodes an overflow error will occur.
' Then note that every node needs at least one connection, although logically 2 is the absolute minimum.
' Nodes can be connected to themselves, but this makes no sense so try to avoid it.
' arrConstraints is an array that describes the constraints a node can have in space.
' Every entry describes a single node. It can have the following values:
' Non-UUID string = the point is free to move in space if there is no object that has an ID which is identical to the supplied string. Using zero length strings to make sure nodes are free.
' UUID-String = the point is locked to an object (can be curve, surface or polysurface) by ClosestPoint (ClosestPoint routines are not included in this routine.
' Instead native Rhino 3.x script methods are called. If you intend to run this script on alternate platform you'll have to translate these calls into native methods. Also if other platforms use different object identifiers (instead of strings), you should make further alterations).
' Damping = a real number controlling the damping of the relaxation routine. Numbers higher than one will accelerate the relaxation, likely causing instable results. This however can be combatted by limiting the maximum distance nodes are allowed to travel. Numbers between zero and one will decelerate the relaxation thus preventing bouncing solutions, but more steps will be required to reach a stable solution within tolerance. Negative numbers should be avoided.
' Limit = a real number controlling the distance every node is allowed to move during each iteration.
' Summary will be filled with an array of data regarding the details of the entire relaxation process
' it will be filled with the following values:
' 0 = The sum-total of distances of all node translations
' 1 = The biggest resultant vector in the set
' The return value of the function is an array identical to arrNodes but with different coordinates

Private Function RelaxMeshNodes(ByRef arrNodes, ByRef arrConnections, ByRef arrConstraints,

ByVal Damping, ByVal Limit, ByRef Summary)

Dim i, j
Dim arrV(), resV
Dim newNodes
Dim ptCP

Dim maxResVec, curResVec
Dim sumNodeTrans

maxResVec = 0.0
sumNodeTrans = 0.0

```

newNodes = arrNodes
For i = 0 To UBound(newNodes)
    If arrConstraints(i) <> "FIXED" Then
        Erase arrV
        ReDim arrV(UBound(arrConnections(i)))
        For j = 0 To UBound(arrV)
            arrV(j) = MultiplyVector(Array(newNodes(i),
newNodes(arrConnections(i)(j)(0))), _
arrConnections(i)(j)(1))
        Next
        resV = ResultantVector(arrV)
        resV = MultiplyVector(resV, Damping)
        resV = LimitVector(resV, Limit)

        If Rhi no. IsObject(arrConstraints(i)) Then
            If Rhi no. IsCurve(arrConstraints(i)) Then
                ptCP = Rhi no. EvaluateCurve(arrConstraints(i),
Rhi no. CurveClosestPoint(arrConstraints(i), resV(1)))
            Else If Rhi no. IsBREP(arrConstraints(i)) Then
                ptCP = Rhi no. BrepClosestPoint(arrConstraints(i),
resV(1))(0)
            End If

            If IsArray(ptCP) Then
                newNodes(i) = ptCP
                curResVec = PointPointDistance(newNodes(i),
arrNodes(i))
                If curResVec > maxResVec Then maxResVec =
curResVec
                sumNodeTrans = sumNodeTrans + curResVec
            Else
                newNodes(i) = resV(1)
                curResVec = PointPointDistance(newNodes(i),
arrNodes(i))
                If curResVec > maxResVec Then maxResVec =
curResVec
                sumNodeTrans = sumNodeTrans + curResVec
            End If
        Else
            newNodes(i) = resV(1)
            curResVec = PointPointDistance(newNodes(i),
arrNodes(i))
            If curResVec > maxResVec Then maxResVec = curResVec
            sumNodeTrans = sumNodeTrans + curResVec
        End If
    End If
Next

Summary = Array(sumNodeTrans, maxResVec)
RelaxMeshNodes = newNodes
End Function

Private Function PointPointDistance(ByRef arrPoint1, ByRef arrPoint2)
    PointPointDistance = (arrPoint1(0)-arrPoint2(0)) * (arrPoint1(0)-arrPoint2(0)) + _
(arrPoint1(1)-arrPoint2(1)) *
(arrPoint1(1)-arrPoint2(1)) + _
(arrPoint1(2)-arrPoint2(2)) *
(arrPoint1(2)-arrPoint2(2))
    PointPointDistance = Sqr(PointPointDistance)
End Function

' This function calculates the resultant vector of an array of vectors.
' The starting point of the first vector in the array will be used as grip
' for the resultant vector, if an invalid vector array is passed an error will occur
Private Function ResultantVector(ByRef arrVectors)
    Dim i
    Dim di vX, di vY, di vZ
    Dim resX, resY, resZ
    Dim ptT(2), ptH(2)

    resX = 0.0
    resY = 0.0
    resZ = 0.0
    For i = 0 To UBound(arrVectors)
        di vX = arrVectors(i)(1)(0) - arrVectors(i)(0)(0)
        di vY = arrVectors(i)(1)(1) - arrVectors(i)(0)(1)
        di vZ = arrVectors(i)(1)(2) - arrVectors(i)(0)(2)
        resX = resX + di vX
        resY = resY + di vY
        resZ = resZ + di vZ
    Next
    ptT(0) = arrVectors(0)(0)(0)
    ptT(1) = arrVectors(0)(0)(1)

```

```

    ptT(2) = arrVectors(0)(0)(2)
    ptH(0) = ptT(0) + resX
    ptH(1) = ptT(1) + resY
    ptH(2) = ptT(2) + resZ

    ResultantVector = Array(ptT, ptH)
End Function

' This function will limit a vectorelength to a certain value
' vectors smaller than the specified length will remain untouched.
' Note that it is unhealthy to use negative numbers or zero for dblMaxLength
Private Function LimitVector(ByVal vecIn, ByVal dblLimitation)
    Dim vecOut(1)
    Dim l

    vecOut(0) = vecIn(0)
    vecOut(1) = vecIn(1)
    l = PointPointDistance(vecIn(0), vecIn(1))

    If l > dblLimitation Then
        vecOut(1)(0) = vecOut(0)(0) + ((vecOut(1)(0)-vecOut(0)(0)) / l) *
dblLimitation
        vecOut(1)(1) = vecOut(0)(1) + ((vecOut(1)(1)-vecOut(0)(1)) / l) *
dblLimitation
        vecOut(1)(2) = vecOut(0)(2) + ((vecOut(1)(2)-vecOut(0)(2)) / l) *
dblLimitation
    End If
    LimitVector = vecOut
End Function

' This function will multiply a vectorelength by a specified factor.
Private Function MultiplyVector(ByVal vecIn, ByVal dblFactor)
    Dim vecOut(1)
    vecOut(0) = vecIn(0)
    vecOut(1) = vecIn(1)
    vecOut(1)(0) = vecOut(0)(0) + (vecOut(1)(0) - vecOut(0)(0)) * dblFactor
    vecOut(1)(1) = vecOut(0)(1) + (vecOut(1)(1) - vecOut(0)(1)) * dblFactor
    vecOut(1)(2) = vecOut(0)(2) + (vecOut(1)(2) - vecOut(0)(2)) * dblFactor
    MultiplyVector = vecOut
End Function

' -----
' -----
' -----
' -----
' -----Thus ends the mesh relaxation source
code...-----
' -----Below you will find an subroutine which implements the above
routine-----
' -----and also adds interface code. (This subroutine requires
RhinoCeros 3.x)-----
' -----
' -----
' -----
' -----

' This function translates a Rhino Nurbs Surface definition into an FDM Mesh definition
' idSurface = The ID of the surface to convert. If the ID is invalid, Null will be returned
' If the surface is a non-compatible FDM object, Null will be
returned
' Surface that are closed or periodic in the v-direction should not
be used. Trims will be ignored.
' arrNodes = Points to a Variant variable that will receive the Mesh Node data
' arrConnections = Points to a Variant variable that will receive the Mesh Connectivity data
' arrConstraints = Points to a Variant variable that will receive the Mesh Constraint data
' WarpTension = A double representing the tension of the fabric in Warp direction (U)
' WeftTension = A double representing the tension of the fabric in Weft direction (V)
' EdgeTension = An Array containing tension factor for all 4 surface edges 0=umin; 1=umax;
2=vmin; 3=vmax
' EdgeLinks = An Array containing constraint information for all four edges. Values can be:
' "FIXED" = Fixed edge
' objID = Linked edge
' <anything else> = Free edge. String that are not "FIXED" and do not represent an
object are automatically treated as free.
' Use "FREE" to skip a few steps in the relaxation
loop
' TipLinks = An Array containing constraint information for all four tips. Values can be
identical to EdgeLinks
' 0=umin, vmin; 1=umax, vmin; 2=umin, vmax; 3=umax, vmax
' If the function succeeds True will be returned, if the function fails Null will be
returned.
```



```

Private Function NURBS2FDM(ByVal idSurface, ByRef arrNodes, ByRef arrFaces, _
ByRef arrConnections, ByRef
arrConstraints, _
ByVal WarpTension, ByVal WeftTension,
ByVal EdgeTension, _
ByVal EdgeLinks, ByVal TipLinks)
    NURBS2FDM = Null

    Dim u, v, N, M
    Dim arrN(), arrF(), arrC(), arrL()
    Dim rhPoints
    Dim uCount, vCount
    Dim uDegree, vDegree
    Dim strChoice

    rhPoints = Rhino.SurfacePoints(idSurface)
    uCount = Rhino.SurfacePointCount(idSurface)(0)
    vCount = Rhino.SurfacePointCount(idSurface)(1)
    uDegree = Rhino.SurfaceDegree(idSurface, 0)
    vDegree = Rhino.SurfaceDegree(idSurface, 1)

    If Rhino.IsSurfacePeriodic(idSurface, 0) Then 'periodic in u-direction
        'Set mesh nodes
        ReDim arrN(vCount * (uCount - uDegree) - 1)
        N = 0
        For u = 0 To uCount - uDegree - 1 'pri razeni rhPoints do arrN
            For v = 0 To vCount - 1
                arrN(N) = rhPoints((u * vCount) + v)
                N = N + 1
            Next
        Next

        ReDim arrF((vCount - 1) * (uCount - uDegree) - 1)
        ReDim arrC(UBound(arrN))
        ReDim arrL(UBound(arrN))
        'Set mesh faces
        N = 0
        For u = 0 To uCount - uDegree - 1
            For v = 0 To vCount - 2
                arrF(N) = Array((u * vCount) + v,
((u * vCount) + v + vCount) Mod ((uCount - uDegree) * vCount),
((u * vCount) + v + vCount + 1) Mod
((uCount - uDegree) * vCount), _
(u * vCount) + v + 1)
                N = N + 1
            Next
        Next

        'Set fabric
        For u = 0 To uCount - uDegree - 1
            For v = 1 To vCount - 2
                N = (u * vCount) + v
                arrC(N) = Array(Array(N + 1, WeftTension), _
Array(N - 1, WeftTension), _
Array(((N - vCount) + ((uCount - uDegree) * vCount)) Mod ((uCount - uDegree) * vCount), WarpTension), _
Array((N + vCount) Mod
((uCount - uDegree) * vCount), WarpTension))
                arrL(N) = "FREE"
            Next
        Next

        'Set u-border edges
        For u = 0 To (uCount - uDegree - 1) * vCount Step vCount
            N = u
            arrC(N) = Array(Array((N + vCount) Mod ((uCount - uDegree) * vCount),
EdgeTension(0)), _
Array((N - vCount + ((uCount - uDegree) * vCount)) Mod ((uCount - uDegree) * vCount), EdgeTension(0)),
_
Array(N + 1, WeftTension))
            arrL(N) = EdgeLinks(0)

            N = u + vCount - 1
            arrC(N) = Array(Array((N + vCount + ((uCount - uDegree) * vCount)) Mod
((uCount - uDegree) * vCount), EdgeTension(1)), _
Array((N - vCount + ((uCount - uDegree) * vCount)) Mod ((uCount - uDegree) * vCount), EdgeTension(1)),
_
Array(N - 1, WeftTension))
            arrL(N) = EdgeLinks(1)
        Next
    Else If Rhino.IsSurfaceClosed(idSurface, 0) Then 'closed in u direction
        'Set mesh nodes

```

```

ReDim arrN(vCount * (uCount-1) - 1)
N = 0
For u = 0 To uCount - 2
For v = 0 To vCount - 1
    arrN(N) = rhPoints((u*vCount)+v)
    N = N+1
Next
Next

ReDim arrF((vCount-1)*(uCount-1)-1)
ReDim arrC(UBound(arrN))
ReDim arrL(UBound(arrN))
' Set mesh faces
N = 0
For u = 0 To uCount-2
For v = 0 To vCount-2
    arrF(N) = Array((u*vCount)+v, _
((uCount-1)*vCount), _
((uCount-1)*vCount), _
((u*vCount)+v+vCount) Mod
((u*vCount)+v+vCount+1) Mod
(u*vCount)+v+1)
    N = N+1
Next
Next

' Set fabric
For u = 0 To uCount-2
For v = 1 To vCount-2
    N = (u*vCount)+v
    arrC(N) = Array(Array(N+1, WeftTension), _
Array(N-1, WeftTension), _
Array(((N-vCount)+((uCount-1)*vCount)) Mod ((uCount-1)*vCount), WarpTension), _
Array((uCount-1)*vCount), WarpTension))
    arrL(N) = "FREE"
Next
Next

' Set u-border edges
For u = 0 To (uCount-2)*vCount Step vCount
    N = u
    arrC(N) = Array(Array((N+vCount) Mod ((uCount-1)*vCount),
EdgeTension(0)), _
Array((N-vCount+((uCount-1)*vCount)) Mod ((uCount-1)*vCount), EdgeTension(0)), _
Array(N+1, WeftTension))
    arrL(N) = EdgeLinks(0)
    N = u+vCount-1
    arrC(N) = Array(Array((N+vCount+((uCount-1)*vCount)) Mod
((uCount-1)*vCount), EdgeTension(1)), _
Array((N-vCount+((uCount-1)*vCount)) Mod ((uCount-1)*vCount), EdgeTension(1)), _
Array(N-1, WeftTension))
    arrL(N) = EdgeLinks(1)
Next
Else
' rectangular patch
' Set mesh nodes
ReDim arrN(UBound(rhPoints))
For N = 0 To UBound(rhPoints)
    arrN(N) = rhPoints(N)
Next
' Set mesh faces
ReDim arrF((uCount-1)*(vCount-1)-1)
N = 0
For u = 0 To uCount-2
For v = 0 To vCount-2
    arrF(N) = Array((u*vCount)+v, (u*vCount)+v+vCount,
(u*vCount)+v+1+vCount, (u*vCount)+v+1)
    N = N+1
Next
Next
' Set connections and link-constraints
ReDim arrC(UBound(arrN))
ReDim arrL(UBound(arrN))
For N = 0 To UBound(arrN)
    arrC(N) = Array(Array(N+1, WeftTension), Array(N-1, WeftTension), _
Array(N-vCount, WarpTension),
Array(N+vCount, WarpTension))
    arrL(N) = "FREE"
Next
' Set u-border edge cables

```

```

        For N = 0 To vCount*(uCount-1) Step vCount
            arrC(N) = Array(Array(N+vCount, EdgeTension(0)), Array(N-vCount,
EdgeTension(0)), Array(N+1, WeftTension))
            arrL(N) = EdgeLinks(0)
            M = N+vCount-1
            arrC(M) = Array(Array(M+vCount, EdgeTension(1)), Array(M-vCount,
EdgeTension(1)), Array(M-1, WeftTension))
            arrL(M) = EdgeLinks(1)
        Next
        'Set v-border edge cables
        For N = 0 To vCount-1
            arrC(N) = Array(Array(N+vCount, WarpTension), Array(N+1,
EdgeTension(2)), Array(N-1, EdgeTension(2)))
            arrL(N) = EdgeLinks(2)
            M = N + (uCount-1)*vCount
            arrC(M) = Array(Array(M-vCount, WarpTension), Array(M-1,
EdgeTension(3)), Array(M+1, EdgeTension(3)))
            arrL(M) = EdgeLinks(3)
        Next
        'Set tips
        arrC(0) = Array(Array(1, EdgeTension(2)), Array(uCount, EdgeTension(0)))
        arrL(0) = TipLinks(0)
        arrC(vCount-1) = Array(Array(vCount-2, EdgeTension(2)), Array(2*vCount-1,
EdgeTension(1)))
        arrL(vCount-1) = TipLinks(2)
        arrC(vCount*(uCount-1)) = Array(Array(vCount*(uCount-1)+1, EdgeTension(3)),
Array(vCount*(uCount-2), EdgeTension(0)))
        arrL(vCount*(uCount-1)) = TipLinks(1)
        arrC(vCount*uCount-1) = Array(Array(vCount*uCount-2, EdgeTension(3)),
Array(vCount*(uCount-1)-1, EdgeTension(1)))
        arrL(vCount*uCount-1) = TipLinks(3)
    End If

    arrNodes = arrN
    arrFaces = arrF
    arrConnections = arrC
    arrConstraints = arrL
    NURBS2FDM = True
End Function

Private Function RecreateFDMSurface(idSurface, arrNodes)
    Dim newNodes()
    Dim u, v, N, index
    Dim Nu, Nv, Du

    Nu = Rhino.SurfacePointCount(idSurface)(0)
    Nv = Rhino.SurfacePointCount(idSurface)(1)
    Du = Rhino.SurfaceDegree(idSurface, 0)
    ReDim newNodes(Nu*Nv-1)

    If Rhino.IsSurfacePeriodic(idSurface, 0) Then
        N = 0
        For u = 0 To Nu-1
            For v = 0 To Nv-1
                index = (u*Nv+v) Mod ((Nu-Du)*Nv)
                newNodes(N) = arrNodes(index)
                N = N+1
            Next
        Next
    Else If Rhino.IsSurfaceClosed(idSurface, 0) Then
        N = 0
        For u = 0 To Nu-1
            For v = 0 To Nv-1
                index = (u*Nv+v) Mod ((Nu-1)*Nv)
                newNodes(N) = arrNodes(index)
                N = N+1
            Next
        Next
    Else
        For N = 0 To UBound(arrNodes)
            newNodes(N) = arrNodes(N)
        Next
    End If
    RecreateFDMSurface = Rhino.AddNurbsSurface(Rhino.SurfacePointCount(idSurface), _
newNodes, _
Rhino.SurfaceKnots(idSurface)(0), _
Rhino.SurfaceKnots(idSurface)(1), _
Rhino.SurfaceDegree(idSurface), _
Rhino.SurfaceWeights(idSurface))
End Function

```

```

' This function displays a message at the command line, blinks it with a red background
colour and then offers a set of options.
' Might be handy to use for feedback...
Private Function FlashMessage(strMessage, arrOptions)
    Dim rgbCurrent, rgbFlash
    Dim lngFlashPause
    lngFlashPause = 175
    rgbCurrent = Rhino.AppearanceColor(12)
    rgbFlash = RGB(255, 0, 0)

    Rhino.Prompt strMessage
    Rhino.AppearanceColor 12, rgbFlash
    Rhino.Sleep lngFlashPause
    Rhino.AppearanceColor 12, rgbCurrent
    Rhino.Sleep lngFlashPause
    Rhino.AppearanceColor 12, rgbFlash
    Rhino.Sleep lngFlashPause
    Rhino.AppearanceColor 12, rgbCurrent
    FlashMessage = Rhino.GetString(strMessage, "", arrOptions)
End Function

' This function creates 4 textdots at the surface corners and prompts the user to pick some.
' You have to remove the objects again by calling the function CLEARPREVIEW afterwards.
' The return value is an array that contains the indices of the edges or NULL on
error/nopick
' 0=umin, vmin 1=umax, vmin 2=umin, vmax 3=umax, vmax
Private Function GetTips(ByVal idSurface, ByVal strPrompt)
    GetTips = Null
    Dim arrDots(3), selDots
    Dim srfDomain, i, j
    Dim arrAll(), S

    If Rhino.IsSurfacePeriodic(idSurface, 0) Or Rhino.IsSurfacePeriodic(idSurface, 1)
Or _
        Rhino.IsSurfaceClosed(idSurface, 0) Or Rhino.IsSurfaceClosed(idSurface, 1) Then
        FlashMessage "The surface has no corners... You'll need to fix or link the
edges instead.", Array("OK")
    Else
        srfDomain = Array(Rhino.SurfaceDomain(idSurface, 0),
Rhino.SurfaceDomain(idSurface, 1))
        arrDots(0) = Rhino.AddTextDot("A", Rhino.EvaluateSurface(idSurface,
Array(srfDomain(0)(0), srfDomain(1)(0))))
        arrDots(1) = Rhino.AddTextDot("B", Rhino.EvaluateSurface(idSurface,
Array(srfDomain(0)(1), srfDomain(1)(0))))
        arrDots(2) = Rhino.AddTextDot("C", Rhino.EvaluateSurface(idSurface,
Array(srfDomain(0)(0), srfDomain(1)(1))))
        arrDots(3) = Rhino.AddTextDot("D", Rhino.EvaluateSurface(idSurface,
Array(srfDomain(0)(1), srfDomain(1)(1))))
        Rhino.ObjectName arrDots, "Gel f l i n g R e l a x a t i o n P r e v i e w O b j e c t s"

        selDots = Rhino.GetObjects(strPrompt, 0, False, False, True, arrDots)
        If IsNull(selDots) Then Exit Function
        S = 0
        For i = 0 To UBound(selDots)
            For j = 0 To 3
                If selDots(i) = arrDots(j) Then
                    ReDim Preserve arrAll(S)
                    arrAll(S) = j
                    S = S+1
                End If
            Next
        Next
        GetTips = arrAll
    End If
End Function

' This function is a wrapper for the Rhino.ExtractIsoCurve method. Differences:
' - it returns only a single ID instead of an array.
' - if the extracted edge is very short a point will be drawn instead.
Private Function ExtractIsoCurveWrapper(ByVal idSurface, ByVal arrParam, ByVal Direction)
    Dim varResult
    varResult = Rhino.ExtractIsoCurve(idSurface, arrParam, Direction)
    If IsNull(varResult) Then
        ExtractIsoCurveWrapper = Rhino.AddPoint(Rhino.EvaluateSurface(idSurface,
arrParam))
    If IsNull(ExtractIsoCurveWrapper) Then ExtractIsoCurveWrapper =
"null-object"
    Else If Rhino.CurveLength(varResult(0)) <= Rhino.UnitsAbsoluteTolerance Then
        Rhino.DeleteObject varResult(0)
        ExtractIsoCurveWrapper = Rhino.AddPoint(Rhino.EvaluateSurface(idSurface,
arrParam))
    If IsNull(ExtractIsoCurveWrapper) Then ExtractIsoCurveWrapper =
"null-object"
    Else

```



```

        ExtractIsoCurveWrapper = varResult(0)
    End If
End Function

' This function copies the edges of a surface and prompts the user to pick some.
' The return value is an array with edge indices or NULL on error/nopick
' 0=umin 1=umax 2=vmin 3=vmax
Private Function GetEdges(ByVal idSurface, ByVal strPrompt)
    GetEdges = Null
    Dim arrEdges(3), selEdges
    Dim srfDomain(1), midDomain(1)
    Dim i, j
    Dim arrAll(), S

    srfDomain(0) = Rhino.SurfaceDomain(idSurface, 0)
    midDomain(0) = (srfDomain(0)(0) + srfDomain(0)(1))/2
    srfDomain(1) = Rhino.SurfaceDomain(idSurface, 1)
    midDomain(1) = (srfDomain(1)(0) + srfDomain(1)(1))/2
    If Rhino.IsSurfacePeriodic(idSurface, 1) Or Rhino.IsSurfaceClosed(idSurface, 1)
Then
        arrEdges(2) = ExtractIsoCurveWrapper(idSurface, Array(srfDomain(0)(0),
midDomain(1)), 1)
        arrEdges(3) = ExtractIsoCurveWrapper(idSurface, Array(srfDomain(0)(1),
midDomain(1)), 1)
    Else If Rhino.IsSurfacePeriodic(idSurface, 0) Or Rhino.IsSurfaceClosed(idSurface, 0)
Then
        arrEdges(0) = ExtractIsoCurveWrapper(idSurface, Array(midDomain(0),
srfDomain(1)(0)), 0)
        arrEdges(1) = ExtractIsoCurveWrapper(idSurface, Array(midDomain(0),
srfDomain(1)(1)), 0)
    Else
        arrEdges(0) = ExtractIsoCurveWrapper(idSurface, Array(midDomain(0),
srfDomain(1)(0)), 0)
        arrEdges(1) = ExtractIsoCurveWrapper(idSurface, Array(midDomain(0),
srfDomain(1)(1)), 0)
        arrEdges(2) = ExtractIsoCurveWrapper(idSurface, Array(srfDomain(0)(0),
midDomain(1)), 1)
        arrEdges(3) = ExtractIsoCurveWrapper(idSurface, Array(srfDomain(0)(1),
midDomain(1)), 1)
    End If

    For i = 0 To 3
        If IsNull(arrEdges(i)) Or IsEmpty(arrEdges(i)) Then arrEdges(i) =
"null-object"
    Next
    Rhino.ObjectName arrEdges, "Gel fli ngRel axati onPrevi ewObj ects"

    selEdges = Rhino.GetObjects(strPrompt, 0, False, False, True, arrEdges)
    If IsNull(selEdges) Then Exit Function
    S = 0
    For i = 0 To UBound(selEdges)
        For j = 0 To 3
            If selEdges(i) = arrEdges(j) Then
                ReDim Preserve arrAll(S)
                arrAll(S) = j
                S = S+1
            End If
        Next
    Next
    GetEdges = arrAll
End Function

' This function draws a set of curves on a surface, either in U/warp or V/weft
' You have to call the CLEARPREVIEW function to remove the curves again afterwards.
Private Sub DrawWarpWeftPreview(ByVal strSurface, ByVal intDirection, ByVal dblDensity)
    Dim uDomain, vDomain
    Dim i, u, v
    Dim crvAdd

    uDomain = Rhino.SurfaceDomain(strSurface, 0)
    vDomain = Rhino.SurfaceDomain(strSurface, 1)

    Rhino.EnableRedraw False
    For i = 0 To 1 Step 1/dblDensity
        u = uDomain(0) + (uDomain(1)-uDomain(0))*i
        v = vDomain(0) + (vDomain(1)-vDomain(0))*i
        crvAdd = Rhino.ExtractIsoCurve(strSurface, Array(u,v), intDirection)
        Rhino.ObjectName crvAdd, "Gel fli ngRel axati onPrevi ewObj ects"
    Next
    Rhino.EnableRedraw True
End Sub

' This function removes all objects from the document that have been created by functions in
this script.
' Basically it removes all objects that are named "Gel fli ngRel axati onPrevi ewObj ects"

```

```

Private Sub ClearPreview()
    Dim allPreviewObjects
    allPreviewObjects = Rhino.ObjectsByName("GelFlingRelaxationPreviewObjects", False)
    If IsNull(allPreviewObjects) Then Exit Sub
    Rhino.DeleteObjects allPreviewObjects
End Sub

' This function checks a surface to see whether it is suitable for FDM relaxation.
' If it turns out not to be, then it will prompt the user for action.
' If the surface could not be or was not fixed NULL will be returned.
' If the surface is acceptable or was fixed the new ID will be returned
Private Function CheckNURBSSurface(ByVal idSurface)
    CheckNURBSSurface = Null
    Dim strResult
    Dim idNewSurface

    idNewSurface = idSurface

    If Not Rhino.IsSurface(idNewSurface) Then Exit Function
    If (Rhino.IsSurfaceClosed(idSurface, 0) Or Rhino.IsSurfacePeriodic(idSurface, 0)) And
    (Rhino.IsSurfaceClosed(idSurface, 1) Or Rhino.IsSurfacePeriodic(idSurface, 1)) Then
        strResult = FlashMessage("The surface you selected has no free edges. Relaxation is not possible.", Array("OK"))
        Exit Function
    End If

    If Rhino.SurfacePointCount(idNewSurface)(0) <= 3 Or
    Rhino.SurfacePointCount(idSurface)(1) <= 3 Then
        strResult = FlashMessage("The surface you selected does not have sufficient grips.", Array("OK"))
        Exit Function
    End If

    If Rhino.IsSurfaceRational(idSurface) Then
        strResult = FlashMessage("The surface you selected is rational. Weighted grips will not translate correctly, what would you like to do?", Array("Unweight", "Ignore", "Abort"))
        If IsNull(strResult) Then Exit Function
        Select Case UCase(strResult)
        Case "UNWEIGHT"
            idNewSurface =
            Rhino.AddNurbsSurface(Rhino.SurfacePointCount(idNewSurface), _
            Rhino.SurfacePoints(idNewSurface), _
            Rhino.SurfaceKnots(idNewSurface)(0), _
            Rhino.SurfaceKnots(idNewSurface)(1), _
            Rhino.SurfaceDegree(idNewSurface))
            Rhino.DeleteObject idSurface
        Case "IGNORE"
            ' Do not take any action
        Case Else
            Exit Function
        End Select
    End If

    If Rhino.IsSurfaceTrimmed(idNewSurface) Then
        strResult = FlashMessage("The surface you selected is trimmed. Trims are not supported, what would you like to do?", Array("Untrim", "Ignore", "Abort"))
        If IsNull(strResult) Then Exit Function
        Select Case UCase(strResult)
        Case "UNTRIM"
            idNewSurface =
            Rhino.AddNurbsSurface(Rhino.SurfacePointCount(idNewSurface), _
            Rhino.SurfacePoints(idNewSurface), _
            Rhino.SurfaceKnots(idNewSurface)(0), _
            Rhino.SurfaceKnots(idNewSurface)(1), _
            Rhino.SurfaceDegree(idNewSurface), _
            Rhino.SurfaceWeights(idNewSurface))
            Rhino.DeleteObject idSurface
        Case "IGNORE"
            ' Do not take any action
        Case Else
            Exit Function
        End Select
    End If

```

```

        If Rhino.IsSurfacePeriodic(idNewSurface, 1) Then
            strResult = FlashMessage("Only u-periodic surfaces are supported, what  
would you like to do?", Array("Swap_UV_Directions", "Abort"))
            If IsNull(strResult) Then Exit Function
            If UCase(strResult) = "ABORT" Then Exit Function
            Rhino.EnableRedraw vbFalse
            Rhino.UnselectAllObjects
            Rhino.SelectObject idNewSurface
            Rhino.Command "-_Dir _SwapUV _Enter", vbFalse
            Rhino.EnableRedraw vbTrue
        End If

        If Rhino.IsSurfaceClosed(idNewSurface, 1) Then
            strResult = FlashMessage("Only u-continues surfaces are supported, what  
would you like to do?", Array("Swap_UV_Directions", "Abort"))
            If IsNull(strResult) Then Exit Function
            If UCase(strResult) = "ABORT" Then Exit Function
            Rhino.EnableRedraw vbFalse
            Rhino.UnselectAllObjects
            Rhino.SelectObject idNewSurface
            Rhino.Command "-_Dir _SwapUV _Enter", vbFalse
            Rhino.EnableRedraw vbTrue
        End If

        CheckNURBSSurface = idNewSurface
    End Function

' This is it. The main sub...
Private Sub RelaxNURBS_Surface()
    Dim arrNodes, arrFaces, arrConnections, arrConstraints
    Dim EdgeTension(3), WarpTension, WeftTension
    Dim EdgeLink(3), TipLink(3)
    Dim strSurfaceID, newSurfaceID
    Dim MeshID
    Dim DampingFactor, Limitation, PreflightLoops, AccuracyMark, LimitationDecay
    Dim Summary
    Dim srfBBox, Diagonal
    Dim RunAsDeveloper, DestinationFolder, FramePrefix, FrameFormat, FrameDim(1)

    ' Set to TRUE to save an animation sequence of the relaxation process.
    RunAsDeveloper = False

    strSurfaceID = Rhino.GetObject("Select a surface for relaxation", 8, True, True)
    If IsNull(strSurfaceID) Then Exit Sub
    strSurfaceID = CheckNURBSSurface(strSurfaceID)
    If IsNull(strSurfaceID) Then
        Rhino.Print "Relaxation process aborted due to improper geometry..."
        Exit Sub
    End If

    Rhino.SurfaceIsoCurveDensity strSurfaceID, -1
    If Not IsNull(Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"LastOperation")) Then
        EdgeTension(0) = Cdbl(Rhino.GetObjectData(strSurfaceID,  
"Gefling_Relaxation", "EdgeTension0"))
        EdgeTension(1) = Cdbl(Rhino.GetObjectData(strSurfaceID,  
"Gefling_Relaxation", "EdgeTension1"))
        EdgeTension(2) = Cdbl(Rhino.GetObjectData(strSurfaceID,  
"Gefling_Relaxation", "EdgeTension2"))
        EdgeTension(3) = Cdbl(Rhino.GetObjectData(strSurfaceID,  
"Gefling_Relaxation", "EdgeTension3"))
        WarpTension = Cdbl(Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"WarpTension"))
        WeftTension = Cdbl(Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"WeftTension"))
        EdgeLink(0) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"EdgeLink0")
        EdgeLink(1) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"EdgeLink1")
        EdgeLink(2) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"EdgeLink2")
        EdgeLink(3) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"EdgeLink3")
        TipLink(0) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"TipLink0")
        TipLink(1) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"TipLink1")
        TipLink(2) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"TipLink2")
        TipLink(3) = Rhino.GetObjectData(strSurfaceID, "Gefling_Relaxation",  
"TipLink3")
    Else
        WarpTension = 1.0
        WeftTension = 1.0
    End If

```

```

        For i = 0 To 3
            EdgeTension(i) = 10.0
            EdgeLink(i) = "FREE"
            TipLink(i) = "FIXED"
        Next
        If Rhino.IsSurfacePeriodic(strSurfaceID, 0) Or
Rhino.IsSurfaceClosed(strSurfaceID, 0) Then
            For i = 0 To 3
                EdgeLink(i) = "FIXED"
            Next
        End If
    End If

    Rhino.Prompt "Loading geometry... please wait"
    Summary = NURBS2FDM(strSurfaceID, arrNodes, arrFaces, arrConnections,
arrConstraints, _
                                WarpTension, WeftTension, EdgeTension,
EdgeLink, TipLink)
    If IsNull(Summary) Then
        MsgBox "Error in building FDM mesh.", vbOkOnly Or vbCritical, "Relaxation
error"
        Exit Sub
    End If

    Dim arrOptions, strResult
    Dim intObject, i, N
    Dim idParent
    Do
        If RunAsDeveloper Then
            arrOptions = Array("Tension_Factors", "Geometry_Links", "Summary",
"Relax", "Relax_advanced", "Quit")
        Else
            arrOptions = Array("Tension_Factors", "Geometry_Links", "Relax",
"Quit")
        End If
        strResult = Rhino.GetString("Relaxation settings", "Relax", arrOptions)
        If IsNull(strResult) Then Exit Sub
        Select Case Left(UCase(strResult), 1)
            Case "M"
                If UCase(strResult) = "MELLON" Then RunAsDeveloper = True
            Case "T"
                Do
                    arrOptions = Array("Warp", "Weft", "Edges", "Return")
                    strResult = Rhino.GetString("Fabric tension factor
properties", "Return", arrOptions)
                    If IsNull(strResult) Then strResult = "Return"
                    Select Case Left(UCase(strResult), 2)
                        Case "WA"
                            DrawWarpWeftPreview strSurfaceID, 0, 30
                            strResult = Rhino.GetReal("Specify a new fabric
warp tension", WarpTension, 0.01, 1000)
                            If Not IsNull(strResult) Then
                                WarpTension = CDBl(strResult)
                            End If
                            ClearPreview
                        Case "WE"
                            DrawWarpWeftPreview strSurfaceID, 1, 30
                            strResult = Rhino.GetReal("Specify a new fabric
weft tension", WeftTension, 0.01, 1000)
                            If Not IsNull(strResult) Then
                                WeftTension = CDBl(strResult)
                            End If
                            ClearPreview
                        Case "ED"
                            intObject = GetEdges(strSurfaceID, "Select edges to
set tension")
                            If IsArray(intObject) Then
                                strResult = Rhino.GetReal("Specify a new
edge cable tension", EdgeTension(intObject(0)), 0.01, 1000)
                                If Not IsNull(strResult) Then
                                    For i = 0 To UBound(intObject)
                                        EdgeTension(intObject(i)) =
CDBl(strResult)
                                    Next
                                End If
                            End If
                            ClearPreview
                        Case Else
                            Exit Do
                    End Select
                Loop
            Case "G"
                Do
                    arrOptions = Array("Corners", "Edges", "Return")
                    strResult = Rhino.GetString("Surface geometry linking

```



```

options", "Return", arrOptions)
    If IsNull(strResult) Then strResult = "Return"
    Select Case Left(UCase(strResult), 1)
    Case "C"
        intObject = GetTips(strSurfaceID, "Select corners
to restrain")
        If IsArray(intObject) Then
            arrOptions = Array("Fixed", "Linked")
            strResult = Rhino.GetString("Pick a node
restraint type", "Fixed", arrOptions)
            If Not IsNull(strResult) Then
                Select Case
                Case "LI"
                    idParent =
Rhino.GetObject("Select a guide object", 4+8+16, False, True)
                    If Not IsNull(idParent)
                        For i = 0 To
                            UBound(intObject)
                                TipLink(intObject(i)) = idParent
                            Next
                        End If
                    Case Else
                        For i = 0 To
                            UBound(intObject)
                                TipLink(intObject(i)) = "FIXED"
                            Next
                        End Select
                    End If
                End If
                ClearPreview
                Rhino.UnselectAllObjects
            Case "E"
                intObject = GetEdges(strSurfaceID, "Select edges to
restrain")
                If IsArray(intObject) Then
                    arrOptions = Array("Fixed", "Free",
"Linked")
                    strResult = Rhino.GetString("Edge restraint
type?", "Fixed", arrOptions)
                    If Not IsNull(strResult) Then
                        Select Case
                        Case "FR"
                            For i = 0 To
                                UBound(intObject)
                                    EdgeLink(intObject(i)) = ""
                                Next
                            Case "LI"
                                idParent =
Rhino.GetObject("Select a guide object", 4+8+16, False, False)
                                If Not IsNull(idParent)
                                    For i = 0 To
                                        UBound(intObject)
                                            EdgeLink(intObject(i)) = idParent
                                        Next
                                    End If
                                Case Else
                                    For i = 0 To
                                        UBound(intObject)
                                            EdgeLink(intObject(i)) = "FIXED"
                                        Next
                                    End Select
                                End If
                            End If
                            ClearPreview
                            Rhino.UnselectAllObjects
                        Case Else
                            Exit Do
                        End Select
                    End If
                End If
                Loop
            Case "S"
                N = 0
                For i = 0 To UBound(arrConnections)
                    N = N + UBound(arrConnections(i))+1
                Next
                MsgBox "NURBS surface relaxation data for object: " & vbNewLine & _
                    "[" & strSurfaceID & "]" & vbNewLine & vbNewLine & _

```

```

"Mesh node count; " & UBound(arrNodes)+1 & vbNewLine & _
"Mesh face count; " & UBound(arrFaces)+1 & vbNewLine & _
"Mesh link count; " & N & vbNewLine & _
"Fabric warp-tension; " & WarpTension & "%" & vbNewLine
& _
"Fabric weft-tension; " & WeftTension & "%" & vbNewLine
& _
"Fabric edge-tension(0); " & EdgeTension(0) & "%" (" &
EdgeLink(0) & ")" & vbNewLine & _
"Fabric edge-tension(1); " & EdgeTension(1) & "%" (" &
EdgeLink(1) & ")" & vbNewLine & _
"Fabric edge-tension(2); " & EdgeTension(2) & "%" (" &
EdgeLink(2) & ")" & vbNewLine & _
"Fabric edge-tension(3); " & EdgeTension(3) & "%" (" &
EdgeLink(3) & ")" & vbNewLine & _
"Fabric corner-link(A); " & TipLink(0) & vbNewLine & _
"Fabric corner-link(B); " & TipLink(1) & vbNewLine & _
"Fabric corner-link(C); " & TipLink(2) & vbNewLine & _
"Fabric corner-link(D); " & TipLink(3) & vbNewLine & _
vbNewLine & _
"end of summary... ", vbOkOnly, "FDM-mesh summary"

Case "R"
    srfBBox = Rhino.BoundingBox(strSurfacelD)
    Diagonal = PointPointDistance(srfBBox(0), srfBBox(6))

    If Len(strResult) = 14 Then
        AccuracyMark = Rhino.GetReal("Specify relaxation
tolerance", Rhino.UnitsAbsoluteTolerance, Rhino.UnitsAbsoluteTolerance/100)
        DampingFactor = Rhino.GetReal("Specify a global damping
factor", 1.0, 0.001, 1000.0)
        Limitation = Rhino.GetReal("Specify an initial translation
limitation", Round(Diagonal * 0.05, 2), AccuracyMark, Diagonal*10)
        LimitationDecay = Rhino.GetReal("Specify a translation
limitation decay factor per iteration", 0.95, 0.1, 0.99)
        PreflightLoops = Rhino.GetInteger("Specify the number of
preflight-iterations", 20, 1, 1000)
        If IsNull(AccuracyMark) Then AccuracyMark =
Rhino.UnitsAbsoluteTolerance
        If IsNull(DampingFactor) Then DampingFactor = 1.0
        If IsNull(Limitation) Then Limitation = Diagonal * 0.05
        If IsNull(LimitationDecay) Then LimitationDecay = 0.95
        If IsNull(PreflightLoops) Then PreflightLoops = 20

    Else
        DampingFactor = 1.0
        Limitation = Diagonal * 0.05
        PreflightLoops = 20
        AccuracyMark = Rhino.UnitsAbsoluteTolerance
        LimitationDecay = 0.95

    End If
    Exit Do

Case "Q"
    Exit Sub

Case Else
    Rhino.Print "Unknown command entered... nothing done."
End Select

Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "LastOperation",
Cstr(Now)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeTension0",
Cstr(EdgeTension(0))
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeTension1",
Cstr(EdgeTension(1))
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeTension2",
Cstr(EdgeTension(2))
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeTension3",
Cstr(EdgeTension(3))
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "WarpTension",
Cstr(WarpTension)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "WeftTension",
Cstr(WeftTension)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeLink0",
EdgeLink(0)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeLink1",
EdgeLink(1)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeLink2",
EdgeLink(2)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "EdgeLink3",
EdgeLink(3)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "TipLink0",
TipLink(0)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "TipLink1",
TipLink(1)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "TipLink2",
TipLink(2)
Rhino.SetObjectData strSurfacelD, "Gel fling_Rel axation", "TipLink3",

```

```

TipLink(3)
Loop

Rhino.UnselectAllObjects
Rhino.HideObject strSurfaceID

Summary = NURBS2FDM(strSurfaceID, arrNodes, arrFaces, arrConnections,
arrConstraints, _
WarpTension, WeftTension, EdgeTension,
EdgeLink, TipLink)
If IsNull(Summary) Then
msgBox "The surface could not be translated into an FDM-mesh." & vbNewLine &
_
"The fact that you're seeing this warning means there" & vbNewLine
& _
"is a bug in the script. Please mail your surface file" & vbNewLine
& _
"(*.3dm fileformat) to david@ruten.as", vbOkOnly Or vbCritical,
"Relaxation routine bug"
Exit Sub
End If

Rhino.Command "-_AdvancedDisplay _Wires _ShowMeshWires=Yes _Enter _Enter", vbFalse
MeshID = Rhino.AddMesh(arrNodes, arrFaces)

DestinationFolder = Null
If RunAsDeveloper Then
DestinationFolder = Rhino.BrowseForFolder("Select a folder to save the
animation sequence. Press 'Cancel' to not save animation.", "Animation settings")
If Not IsNull(DestinationFolder) Then
FramePrefix = Rhino.StringBox("Enter a name for the sequence...",
"RelaxationFrame", "Animation settings")
If IsNull(FramePrefix) Then
DestinationFolder = Null
Else
FrameFormat = Rhino.ListBox(Array("bmp", "tga", "pcx",
"jpg", "png", "tif"), "Select a file format", "Animation settings")
If IsNull(FrameFormat) Then
DestinationFolder = Null
Else
Summary = Rhino.PropertyListBox(Array("Animation
width", "Animation height"), _
Array("1500", "1000"), "Enter frame dimensions", "Animation settings")
If IsNull(Summary) Then
DestinationFolder = Null
Else
FrameDim(0) = Summary(0)
FrameDim(1) = Summary(1)
End If
End If

If Not IsNull(DestinationFolder) Then
Rhino.Command "-_TestViewCaptureToFile " & Chr(34) &
DestinationFolder & FramePrefix & "00000." & FrameFormat & Chr(34) & _
" _Width=" & FrameDim(0) & "
_Height=" & FrameDim(1) & " _DrawCPlane=No _DrawWorldAxes=No _Enter", vbFalse
End If
End If

'Analyse current FDM mesh properties
Rhino.Prompt "Performing test relaxation run... please wait"
RelaxMeshNodes arrNodes, arrConnections, arrConstraints, DampingFactor, Limitation,
Summary
Rhino.Print "Initial cumulative node translation: " & Round(Summary(0)) &
Rhino.UnitsSystemName(False, True, True) & _
" Initial largest node vector: " & Round(Summary(1), 2) &
" N"

'Perform preflight relaxation
For i = 1 To PreflightLoops
Rhino.Prompt "Performing preflight relaxation. Step " & i & " of " &
PreflightLoops & "... "
arrNodes = RelaxMeshNodes(arrNodes, arrConnections, arrConstraints,
DampingFactor, Limitation, Summary)
Rhino.EnableRedraw vbFalse
Rhino.DeleteObject MeshID
MeshID = Rhino.AddMesh(arrNodes, arrFaces)
Rhino.EnableRedraw vbTrue

If Not IsNull(DestinationFolder) Then
Rhino.Command "-_TestViewCaptureToFile " & Chr(34) &
DestinationFolder & FramePrefix & Right("00000" & i, 5) & "." & FrameFormat & Chr(34) & _

```

```

        _Height=" & FrameDim(1) & " _DrawCplane=No _DrawWorldAxes=No _Enter", vbFalse
    End If
Next
' Perform damped relaxation
Do
    arrNodes = RelaxMeshNodes(arrNodes, arrConnections, arrConstraints,
DampingFactor, Limitation, Summary)
    Rhino.Prompt "Iteration " & i & _
    Damping: " & Round(DampingFactor*100, 2) & "%
    & _
    Limitation: " & Round(Limitation, 2) &
Rhino.UnitsSystemName(False, True, True)

    Limitation = Limitation * LimitationDecay
    'DampingFactor = DampingFactor * 0.98

    Rhino.EnableRedraw vbFalse
    Rhino.DeleteObject MeshID
    MeshID = Rhino.AddMesh(arrNodes, arrFaces)
    Rhino.EnableRedraw vbTrue

    If Not IsNull(DestinationFolder) Then
        Rhino.Command "-_TestViewCaptureToFile " & Chr(34) &
DestinationFolder & FramePrefix & Right("00000" & i, 5) & ". " & FrameFormat & Chr(34) & _
        _Height=" & FrameDim(1) & " _DrawCplane=No _DrawWorldAxes=No _Enter", vbFalse
    End If

    If Limitation < AccuracyMark Then Exit Do
    i = i+1
Loop

Rhino.EnableRedraw False
newSurfaceID = RecreateFDMSurface(strSurfaceID, arrNodes)
If IsNull(newSurfaceID) Then
    MsgBox "New surface could not be created." & vbNewLine & _
        "The mesh will be preserved instead...", vbOkOnly Or
vbExclamation, "Error!"
    Rhino.ShowObject strSurfaceID
    Rhino.EnableRedraw True
Exit Sub
End If

Rhino.ShowObject strSurfaceID
Rhino.DeleteObject MeshID
Rhino.DeleteObject strSurfaceID
Rhino.SelectObject newSurfaceID

Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "LastOperation",
Cstr(Now)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeTension0",
Cstr(EdgeTension(0))
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeTension1",
Cstr(EdgeTension(1))
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeTension2",
Cstr(EdgeTension(2))
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeTension3",
Cstr(EdgeTension(3))
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "WarpTension",
Cstr(WarpTension)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "WeftTension",
Cstr(WeftTension)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeLink0",
EdgeLink(0)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeLink1",
EdgeLink(1)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeLink2",
EdgeLink(2)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "EdgeLink3",
EdgeLink(3)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "Tiplink0",
Tiplink(0)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "Tiplink1",
Tiplink(1)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "Tiplink2",
Tiplink(2)
Rhino.SetObjectData newSurfaceID, "Gel fling_Relaxation", "Tiplink3",
Tiplink(3)
Rhino.EnableRedraw True
End Sub
RelaxNURBS_Surface

```


Příloha č. 3 – Katalog Pfeifer (str. 28, 33 a 38)

Spiralseil DIN EN 12385 – Edelstahl Spiral Strand DIN EN 12385 – Stainless Steel

Zulassungsnummer
Approval-Number
Z-14.7-411

PE

1 x 19



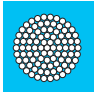
1 x 37



1 x 61



1 x 91



Technische Daten

Material:
gemäß Zulassung Z-14.7-411

Elastizitätsmodul:
 $130 \pm 10 \text{ kN/mm}^2$

Toleranz d_s :
+ 3%

Technical Data

Material:
according Technical Approval Z-14.7-411

Modulus of Elasticity:
 $130 \pm 10 \text{ kN/mm}^2$

Tolerance d_s :
+ 3%



Datenblätter
Data Sheets

Größe size	Charakt. Bruchkraft charact. breaking load $Z_{B,k}$ DIN 18800* kN	Grenzzugkraft limit tension $Z_{R,d}$ DIN 18800 kN	Metall. Querschnitt metallic cross section ca./approx. mm^2	Gewicht weight ca./approx. kg/m	Konstruktion construction	Seil-Nezendurchmesser nomin. strand dia. d_s mm
PE 3	26	16	22	0,2	1 x 19	6,1
PE 5	47	28	38	0,3	1 x 19	8,1
PE 7	73	44	60	0,5	1 x 19	10,1
PE 10	101	61	83	0,7	1 x 19	11,9
PE 15	141	86	117	0,9	1 x 37	14,1
PE 20	195	118	161	1,3	1 x 37	16,6
PE 30	298	180	246	1,9	1 x 37	20,5
PE 45	409	248	338	2,7	1 x 61	24,1
PE 60	578	350	477	3,7	1 x 61	28,6
PE 75	730	442	602	4,7	1 x 91	32,1
PE 100	945	573	780	6,1	1 x 91	36,6

*nach EC 3 = $F_{u,k}$ und nach ASCE 19-96 = S_d
Konstruktionsänderungen vorbehalten
Größere Abmessungen und Zwischengrößen auf Anfrage

*according EC 3 = $F_{u,k}$ and according ASCE 19-96 = S_d
Subject to technical modification
Bigger dimensions and intermediate dimensions upon request

Spiralseil DIN EN 12385 – GALFAN Spiral Strand DIN EN 12385 – GALFAN

Zulassungsnummer
Approval-Number
Z-14.7-413

PG

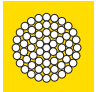
1 x 19



1 x 37



1 x 61



Technische Daten

Material:
unlegierter Qualitätsstahl

Elastizitätsmodul:
 $160 \pm 10 \text{ kN/mm}^2$

Toleranz d_s :
+ 3%

Korrosionsschutz:
GALFAN verzinkt ohne Innenverfüllung

Technical Data

Material:
unalloyed quality steel

Modulus of Elasticity:
 $160 \pm 10 \text{ kN/mm}^2$

Tolerance d_s :
+ 3%

Corrosion Protection:
GALFAN coated without inner filling



Datenblätter
Data Sheets

Größe size	Charakt. Bruchkraft charact. breaking load $Z_{B,k}$ DIN 18800* kN	Grenzzugkraft limit tension $Z_{R,d}$ DIN 18800 kN	Metall. Querschnitt metallic cross section ca./approx. mm^2	Gewicht weight ca./approx. kg/m	Konstruktion construction	Seil-Neiendurchmesser nomin. strand dia. d_s mm
PG 5	59	36	39	0,3	1 x 19	8,1
PG 10	93	56	60	0,5	1 x 19	10,1
PG 15	134	81	87	0,7	1 x 19	12,2
PG 20	181	109	117	0,9	1 x 37	14,1
PG 25	260	158	168	1,3	1 x 37	17,0
PG 40	367	222	237	1,9	1 x 37	20,1
PG 55	537	326	347	2,7	1 x 37	24,4
PG 75	722	438	467	3,7	1 x 37	28,3
PG 90	884	536	572	4,5	1 x 61	31,3
PG 125	1189	721	769	6,1	1 x 61	36,3

*nach EC 3 = $F_{u,k}$ und nach ASCE 19-96 = S_d
Konstruktionsänderungen vorbehalten
Größere Abmessungen und Zwischengrößen auf Anfrage

*according EC 3 = $F_{u,k}$ and according ASCE 19-96 = S_d
Subject to technical modification
Bigger dimensions and intermediate dimensions upon request

Vollverschlossenes Seil – GALFAN Full Locked Cable – GALFAN

Zulassungsnummer
Approval-Number
Z-14.7-413

PV

VVS-1



VVS-2



VVS-3



Technische Daten

Material:
unlegierter Qualitätsstahl

Elastizitätsmodul:
 $160 \pm 10 \text{ kN/mm}^2$

Toleranz d_s :
+ 3%

Korrosionsschutz:
innere Lagen: feuerverzinkt
mit Innenverfüllung
äußere Lagen: GALFAN verzinkt
ohne Innenverfüllung

Technical Data

Material:
unalloyed quality steel

Modulus of Elasticity:
 $160 \pm 10 \text{ kN/mm}^2$

Tolerance d_s :
+ 3%

Corrosion Protection:
inner layers: hot dip galvanized
with inner filling
outer layers: GALFAN coated
without inner filling



Datenblätter
Data Sheets

Größe size	Charakt. Bruchkraft charact. breaking load $Z_{B,k}$ DIN 18800* kN	Grenzzugkraft limit tension $Z_{R,d}$ DIN 18800 kN	Metall. Querschnitt metallic cross section ca./approx. mm ²	Gewicht weight ca./approx. kg/m	Konstruktion construction **	Seil-Nezendurchmesser nomin. strand dia. d_s mm
PV 40	405	245	281	2,4	VVS-1	21
PV 60	621	376	430	3,6	VVS-1	26
PV 90	916	555	634	5,3	VVS-2	31
PV 115	1170	709	808	6,8	VVS-2	35
PV 150	1520	921	1060	8,9	VVS-2	40
PV 195	1930	1170	1340	11,2	VVS-2	45
PV 240	2380	1442	1650	13,8	VVS-2	50
PV 300	3020	1830	2090	17,2	VVS-3	55
PV 360	3590	2176	2490	20,5	VVS-3	60
PV 420	4220	2558	2920	24,1	VVS-3	65
PV 490	4890	2964	3390	27,9	VVS-3	70
PV 560	5620	3406	3890	32,1	VVS-3	75
PV 640	6390	3873	4420	36,4	VVS-3	80
PV 720	7210	4370	4990	41,1	VVS-3	85
PV 810	8090	4903	5600	46,2	VVS-3	90
PV 910	9110	5521	6310	52,0	VVS-3	95
PV 1010	10100	6121	6990	57,6	VVS-3	100
PV 1110	11100	6727	7710	63,5	VVS-3	105
PV 1220	12200	7394	8460	69,7	VVS-3	110
PV 1340	13400	8121	9240	76,2	VVS-3	115
PV 1450	14500	8788	10100	83,2	VVS-3	120
PV 1580	15800	9576	10900	89,8	VVS-3	125
PV 1730	17300	10485	11900	96,7	VVS-3	130
PV 1860	18600	11273	12900	104,8	VVS-3	135
PV 2000	20000	12121	13900	112,9	VVS-3	140

**VVS-1 = 1, VVS-2 = 2, VVS-3 = 3 und mehr Lagen Profilstränge

*nach EC 3 = $F_{t,k}$ und nach ASCE 19-96 = S_d

Unter Vorspannung und / oder Witterungseinflüssen ist der Austritt von Innenverfüllung möglich.
Konstruktionsänderungen vorbehalten

Größere Abmessungen und Zwischengrößen auf Anfrage

**VVS-1 = 1, VVS-2 = 2, VVS-3 = 3 and more layers z-profiled wires

*according EC 3 = $F_{t,k}$ and according ASCE 19-96 = S_d

Due to prestressing and / or differing weather conditions inner filling may escape to the surface.
Subject to technical modifications

Bigger dimensions and intermediate dimensions upon request

Příloha č. 4 – Katalog Mehler (Valmex FR 700, 900, 1000 a 1400)



Technisches Datenblatt Nr.: **1152.2**

Produkt: VALMEX® FR 700 MEHATOP F - type I

Artikel Nr.: 7253 5244

Beschichtung und Ausrüstung		
Beschichtungsart	PVC	
Ausrüstung	aussen PVDF Lack, innen 2x Acryl, mikrobiozid, UV-geschützt, low-wick	
Brennverhalten	BS 7837, California T 19, DIN 4102: B1, NFP 92507: M2	
zu Brennverhalten	stets Gültigkeit der FR-Zulassung prüfen	
Gesamtgewicht	850 g/m ²	EN ISO 2286-2
Reißkraft Kette/Schuß	3000 / 3000 N/50 mm	DIN 53354, DIN EN ISO 1421/V1
Weiterreißfestigkeit(1) Kette/Schuß	300 / 300 N	DIN 53363
Haftfestigkeit (1)	20 N/cm	LB 3.04-1 (Complan)
Kältebeständigkeit	-30 °C	DIN 53361
Wärmebeständigkeit	+70 °C	Complan-Richtlinie
Lichtechtheit	>6 Note	DIN 54004, DIN EN ISO 105 B02
Knickfestigkeit (1)	keine Risse	< 100000 x DIN 53359 A
Trägergewebe		
Material	PES	DIN ISO 2076
Fadenstärke	1100 dtex	DIN ISO 2060
Bindung	L 1/1	
Bemerkungen	ohne Vorbehandlung nach gängigen Verfahren verschweißbar	

Bei den technischen Daten handelt es sich um ca. Werte, die auf Basis von ermittelten Durchschnittswerten erstellt wurden. Aus fertigungstechnischen Gründen sind geringfügige Abweichungen möglich. Diese technischen Angaben entsprechen dem heutigen Stand der Kenntnisse und sollen über unsere Produkte ohne Rechtsverbindlichkeit informieren. Diese Daten gelten für neue Ware.



Technisches Datenblatt Nr.: **1011.4**

Produkt: VALMEX® FR 900 MEHATOP F - type II

Artikel Nr.: 7211 HTL 5246

Beschichtung und Ausrüstung			
Beschichtungsart	PVC		
Ausrüstung	beidseitig PVDF lackiert, mikrobiozid, UV-geschützt, low-wick		
Brennverhalten	BS 7837, California T 19, DIN 4102: B1, NFP 92507: M2		
zu Brennverhalten	stets Gültigkeit der FR-Zulassung prüfen		
Gesamtgewicht	900	g/m²	EN ISO 2286-2
Reißkraft Kette/Schuß	4200 / 4000	N/50 mm	DIN 53354, DIN EN ISO 1421/V1
Weiterreißfestigkeit(1) Kette/Schuß	500 / 450	N	DIN 53363
Haftfestigkeit (1)	25	N/cm	LB 3.04-1 (Complan)
Kältebeständigkeit	-30	°C	DIN 53361
Wärmebeständigkeit	+70	°C	Complan-Richtlinie
Lichtdurchlässigkeit	12 - 15	%	bei 550 nm
Lichteinheit	>6	Note	DIN 54004, DIN EN ISO 105 B02
Knickfestigkeit (1)	keine Risse	< 100000 x	DIN 53359 A
Trärgewebe			
Material	PES		DIN ISO 2076
Fadenstärke	1100	dtex	DIN ISO 2060
Bindung	P 2/2		
Bemerkungen	ohne Vorbehandlung nach gängigen Verfahren verschweißbar		

Bei den technischen Daten handelt es sich um ca. Werte, die auf Basis von ermittelten Durchschnittswerten erstellt wurden. Aus fertigungstechnischen Gründen sind geringfügige Abweichungen möglich. Diese technischen Angaben entsprechen dem heutigen Stand der Kenntnisse und sollen über unsere Produkte ohne Rechtsverbindlichkeit informieren. Diese Daten gelten für neue Ware.



Technisches Datenblatt Nr.: **1018.2**

Produkt: VALMEX® FR 1000 metallic - type III

Artikel Nr.: 7269 5251

Beschichtung und Ausrüstung		
Beschichtungsart	PVC	
Ausrüstung	aussen Metallic-Lack, Rückseite mit Acryllack, mikrobiozid, UV-geschützt	
Brennverhalten	BS 7837, California T 19, DIN 4102: B1	
zu Brennverhalten	stets Gültigkeit der FR-Zulassung prüfen	
Gesamtgewicht	1050 g/m ²	EN ISO 2286-2
Reißkraft Kette/Schuß	6000 / 5500 N/50 mm	DIN 53354, DIN EN ISO 1421/V1
Weiterreißfestigkeit(1) Kette/Schuß	900 / 800 N	DIN 53363
Haftfestigkeit (1)	25 N/cm	LB 3.04-1 (Complan)
Kältebeständigkeit	-30 °C	DIN 53361
Wärmebeständigkeit	+70 °C	Complan-Richtlinie
Lichtehtheit	>6 Note	DIN 54004, DIN EN ISO 105 B02
Knickfestigkeit (1)	keine Risse	< 100000 x DIN 53359 A
Trägergewebe		
Material	PES	DIN ISO 2076
Fadenstärke	1670 dtex	DIN ISO 2060
Bindung	P 2/2	
Bemerkungen	Metallic-Lacke können die Verschweißbarkeit beeinflussen	

Bei den technischen Daten handelt es sich um ca. Werte, die auf Basis von ermittelten Durchschnittswerten erstellt wurden. Aus fertigungstechnischen Gründen sind geringfügige Abweichungen möglich. Diese technischen Angaben entsprechen dem heutigen Stand der Kenntnisse und sollen über unsere Produkte ohne Rechtsverbindlichkeit informieren. Diese Daten gelten für neue Ware.



Technisches Datenblatt Nr.: **1321.2**

Produkt: VALMEX® FR 1400 MEHATOP F - type IV gold

Artikel Nr.: 7270 5251

Beschichtung und Ausrüstung			
Beschichtungsart	PVC		
Ausrüstung	mikrobiozid, UV-geschützt, low-wick		
zu Ausrüstung	Gold-Lack auf Vorderseite, Innenseite Acryl-Lack		
Brennverhalten	BS 7837, DIN 4102: B1		
zu Brennverhalten	stets Gültigkeit der FR-Zulassung prüfen		
Gesamtgewicht	1350	g/m²	EN ISO 2286-2
Reißkraft Kette/Schuß	7500 / 6500	N/50 mm	DIN 53354, DIN EN ISO 1421/V1
Weiterreißfestigkeit(1) Kette/Schuß	1200 / 1200	N	DIN 53363
Haftfestigkeit (1)	25	N/cm	LB 3.04-1 (Complan)
Kältebeständigkeit	-30	°C	DIN 53361
Wärmebeständigkeit	+70	°C	Complan-Richtlinie
Lichtechtheit	>6	Note	DIN 54004, DIN EN ISO 105 B02
Knickfestigkeit (1)	keine Risse	< 100000 x	DIN 53359 A
Trärgewebe			
Material	PES		DIN ISO 2076
Fadenstärke	1670	dtex	DIN ISO 2060
Bindung	P 3/3		
Bemerkungen	Metallic-Lacke können die Verschweißbarkeit beeinflussen, Gold-Pigmente können die Verschweißbarkeit beeinträchtigen. Empfehlungen können angefragt werden.		

Bei den technischen Daten handelt es sich um ca. Werte, die auf Basis von ermittelten Durchschnittswerten erstellt wurden. Aus fertigungstechnischen Gründen sind geringfügige Abweichungen möglich. Diese technischen Angaben entsprechen dem heutigen Stand der Kenntnisse und sollen über unsere Produkte ohne Rechtsverbindlichkeit informieren. Diese Daten gelten für neue Ware.