

Muodon optimointi virtauksen ja rakenteen vuorovaikutusongelmassa

TKK
Mekaniikan laboratorio
Mat-5.002 Mekaniikan erikoistyö
Jani Paavilainen
48661C

31. elokuuta 2001

Sisältö

1 Johdanto	2
2 Ongelman kuvaus	3
3 Matemaattinen tausta	4
3.1 Virtauksen mallintaminen	4
3.2 Rakenteen mallintaminen	5
3.3 Virtaus- ja rakennemallien yhdistäminen	7
3.4 Optimointimenetelmät	8
3.4.1 Toistettu kvadraattinen optimointi	8
3.4.2 Polytooppihaku	9
4 Ratkaisu menetelmät	10
4.1 Käytetyt ohjelmistot	10
4.2 Palkin taipuman laskeminen	10
4.3 Optimoinnin suorittaminen	11
5 Esimerkkiongelmia	13
5.1 Geometria ja alkuperäinen taipuma	13
5.2 Kustannusfunktio	13
5.3 Optimointi	15
6 Esimerkkiongelman numeerinen ratkaiseminen	18
6.1 Mallin mitat, reunaehdot ja materiaaliominaisuudet	18
6.2 Tulokset	18
7 Pohdinnat	24
Viitteet	25
Liitteet	25

1 Johdanto

Virtaus-rakenne vuorovaikutusongelmat ovat hyvin yleisiä insinööritieteissä. Niihin törmää monilla eri tekniikan aloilla, kuten esimerkiksi meri-, lento- ja tuulitekniikassa. Kun tälläisiin kytkettyihin virtaus-rakenne vuorovaikutusongelmiin vielä liitetään optimointi, saadaan haastavia ongelmia. Haasteelliseksi nämä ongelmat tekee monet virtaus-rakenne vuorovaikutuksessa esiintyvät epälineaarisuudet sekä optimointiin liittyvä työläs laskenta.

Tässä erikoistyössä tutkitaan kappaleen muodon optimointia virtauskentässä. Kyseisessä ongelmassa epälineaarisuuksia syntyy sekä virtaavan aineen että kappaleen käyttäytymisestä. Myös virtaavan aineen ja kappaleen muuttuva rajapinta tuo mukanaan ilmiöitä, jotka täytyy laskennassa ottaa huomioon.

Työssä luotiin virtaus-rakenne vuorovaikutusongelmasta malli, joka ratkaistiin valmiilla elementtimenetelmäohjelmistolla. Optimointia varten johdettiin ongelmaan sopiva kustannusfunktio sekä valittiin sopivat rakenteen parametrisointitavat. Itse optimointi suoritettiin käyttäen valmista kaupallista ohjelmistoa.

Aluksi työssä on käyty läpi annettuun ongelmaan liittyvien fysikaalisten ilmiöiden sekä optimoinnin matemaattisia perusteita. Tämän jälkeen on kerrottu hieman ongelman ratkaisussa käytetyistä ohjelmistoista sekä käyty ongelman ratkaisumallit ja -perusteet tarkasti läpi. Sitten on suoritettu esimerkkilaskelmat ja koottu laskennasta saadut tulokset yhteen. Lopuksi on tarkasteltu tuloksia sekä niihin johtaneita menetelmiä.

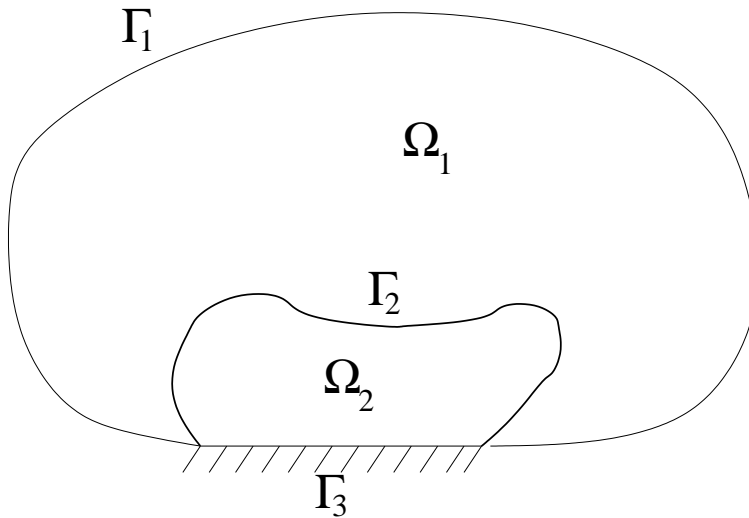
Tämä erikoistyö on tehty osana Tekesin rahoittamaa CSC:n CFD-optimointi projektia.

2 Ongelman kuvaus

Tehtävänä oli etsiä kappaleen optimimuoto virtauksessa siten, että kappaleen maksimitaipuma jatkuvuustilassa olisi mahdollisimman pieni. Optimimuoto tuli löytää siten, että kappaleen tilavuus ja korkeus pysyisivät vakioina.

Yhden taipumaratkaisun laskemisen tavoiteajaksi asetettiin noin kaksi minuuttia, jotta optimointiin ei kuluisi liikaa aikaa. Näin ollen taipumaratkaisun etsimisessä käytettyjen numeeristen menetelmien tuli konvergoida vaadittuun tarkkuuteen kyseisessä ajassa. Tämä rajoitti ongelmasta luodun mallin kokoa sekä virtaavan aineen ja kappaleen ominaisuuksia. Paremmiin käytäntöä vastaavaan tilanteen mallintamiseen olisi vaatinut huomattavan paljon enemmän aikaa taipumaratkaisun etsimiseen ja siten myös optimointiin.

Kuvassa 1 on havainnollistettu ongelman yleistä geometriaa. Siinä kappale Ω_2 on virtauskentässä Ω_1 . Kappaleen optimoitavan muodon määrää reuna Γ_2 , kun taas reunasta Γ_3 kappale on jäykästi kiinnitetty alustaansa. Virtaavana aineena on newtonilainen, kokoonpuristumaton neste. Reuna Γ_1 rajaa sen alueen, jonka sisällä virtausta kuvaavat yhtälöt ratkaistaan.



Kuva 1: Ongelman geometria.

3 Matemaattinen tausta

3.1 Virtauksen mallintaminen

Virtaus mallinnetaan Navier-Stokes yhtälöiden avulla. Nämä yhtälöt johdetaan massan ja liikemäärän säilymislaeista, ks. esim. Hämäläinen ja Järvinen [1]. Massan säilymislaki voidaan kirjoittaa muodossa

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (1)$$

missä $\rho = \rho(x, t)$ on aineen tiheys pisteessä x ajan hetkellä t ja \mathbf{v} on nopeusvektori. Liikemäärän säilymislaki voidaan esittää muodossa

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} = \rho \mathbf{f} + \nabla \cdot \bar{\boldsymbol{\sigma}}, \quad (2)$$

missä ρ on tiheys, t on aika, \mathbf{v} on nopeus, \mathbf{f} on tilavuusvoima ja $\bar{\boldsymbol{\sigma}}$ on jännitystensori. Virtaavan aineen ollessa newtonilaista, on jännitystensori muotoa

$$\bar{\boldsymbol{\sigma}} = -p\mathbf{I} + 2\mu\bar{\boldsymbol{\varepsilon}} - \frac{2}{3}\mu(\nabla \cdot \mathbf{v})\mathbf{I}, \quad (3)$$

missä p on paine, μ on dynaaminen viskositeetti, \mathbf{I} on yksikkötensori ja $\bar{\boldsymbol{\varepsilon}}$ on venymätensori [1]. Lineaarisen venymätensorin komponentit ovat

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \quad (4)$$

Tällöin

$$\nabla \cdot \bar{\boldsymbol{\sigma}} = \nabla \cdot \left[-p\mathbf{I} + 2\mu\bar{\boldsymbol{\varepsilon}} - \frac{2}{3}\mu(\nabla \cdot \mathbf{v})\mathbf{I} \right] \quad (5)$$

$$= -\nabla p + \nabla \cdot (2\mu\bar{\boldsymbol{\varepsilon}}) - \nabla \cdot \left(\frac{2}{3}\mu(\nabla \cdot \mathbf{v}) \right) \quad (6)$$

ja liikemäärän säilymislaki (2) saadaan muotoon [1]

$$\rho \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot (2\mu\bar{\boldsymbol{\varepsilon}}) + \nabla \cdot \left(\frac{2}{3}\mu(\nabla \cdot \mathbf{v}) \right) + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = \rho \mathbf{f}. \quad (7)$$

Kun aine on kokoonpuristumatonta, saadaan massan säilymislaista (1) jatkuvuusyhtälö

$$\nabla \cdot \mathbf{v} = 0. \quad (8)$$

Ottamalla liikeyhtälössä (7) huomioon jatkuvuusyhtälö (8) saadaan yhtälöt

$$\begin{cases} \rho \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot (2\mu\bar{\boldsymbol{\varepsilon}}) + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = \rho \mathbf{f} \\ \nabla \cdot \mathbf{v} = 0, \end{cases} \quad (9)$$

joita kutsutaan Navier-Stokes yhtälöiksi. Viskositeetin ollessa vakio saadaan Navier-Stokes yhtälöt vielä sievennettyä muotoon [1]

$$\begin{cases} \rho \frac{\partial \mathbf{v}}{\partial t} - \mu \nabla^2 \mathbf{v} + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = \rho \mathbf{f} \\ \nabla \cdot \mathbf{v} = 0. \end{cases} \quad (10)$$

Liikkuvan referenssikoordinaatiston tapauksessa materiaaliderivaatta saadaan muotoon

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial\mathbf{v}}{\partial t} + \dot{\mathbf{u}} \cdot \nabla\mathbf{v}, \quad (11)$$

missä $\dot{\mathbf{u}}$ on elementtiverkon muodonmuutosnopeus. Tällöin Navier-Stokes yhtälöt (10) voidaan kirjoittaa ALE (Arbitrary-Lagrangian-Eulerian) muodossa seuraavasti [2]

$$\begin{cases} \rho \left(\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} - (\dot{\mathbf{u}} \cdot \nabla)\mathbf{v} \right) - \mu \nabla^2 \mathbf{v} + \nabla p = \rho \mathbf{f} \\ \nabla \cdot \mathbf{v} = 0. \end{cases} \quad (12)$$

3.2 Rakenteen mallintaminen

Rakenteen mallintamiseen käytetään elastisuusyhtälöitä. Olkoon rakenteen referenssikonfiguraatio $\Omega_R \subset \mathbb{R}^3$ avoin ja rajoitettu. Rakenteen nykyinen deformoitunut konfiguraatio Ω saadaan kuvauksesta

$$\phi : \Omega_R \longrightarrow \mathbb{R}^3, \quad (13)$$

missä $\phi(\mathbf{x})$ vastaa sen pisteen asemaa, joka sijaitsee pisteessä \mathbf{x} referenssikonfiguraatiossa. Kuvaus (13) oletetaan riittävän sileäksi ja

$$\det(\nabla\phi) > 0, \quad (14)$$

missä $\nabla\phi$ on muodonmuutosgradientti [3]. Rakenteen siirtymä \mathbf{u} määritetään yhtälöstä

$$\phi = \mathbf{x} + \mathbf{u}(\mathbf{x}). \quad (15)$$

Määritellään Caychy-Green venymätensori [3]

$$\overline{\overline{C}} = \nabla\phi^T \nabla\phi \quad (16)$$

ja Green-St. Venant venymä [3]

$$\overline{\overline{E}} = \frac{1}{2}(\overline{\overline{C}} - \mathbf{I}). \quad (17)$$

Sijoittamalla yhtälöt (15) ja (16) Green-St. Venantin venymän yhtälöön (17) saadaan kyseinen yhtälö kirjoitettua siirtymien avulla muotoon

$$\overline{\overline{E}} = \frac{1}{2}(\nabla\mathbf{u}^T + \nabla\mathbf{u} + \nabla\mathbf{u}^T \nabla\mathbf{u}), \quad (18)$$

jota kutsutaan Green-Lagrangen venymätensoriksi. Cauchyn jännitystensori $\overline{\overline{T}}$ toteuttaa deformoituneessa konfiguraatiossa Ω tasapainoyhtälön

$$-\operatorname{div} \overline{\overline{T}} = \mathbf{f}, \quad (19)$$

missä

$$\mathbf{f} = \det(\nabla\phi^{-1}) \mathbf{f}_R \quad (20)$$

on tilavuusvoima [3]. Alaindeksi \mathbf{R} viittaa referenssikonfiguraatioon. Piola-Kirchhoffin ensimmäinen jännitystensori

$$\overline{\overline{\mathbf{T}}}_R = \det(\nabla\phi)\overline{\overline{\mathbf{T}}}(\nabla\phi)^{-T} \quad (21)$$

taas toteuttaa tasapainoyhtälön

$$-\operatorname{div}_R \overline{\overline{\mathbf{T}}}_R = \mathbf{f}_R \quad (22)$$

referenssikonfiguraatiossa Ω_R [3]. Koska Piola-Kirchhoffin ensimmäinen jännitystensori (21) ei ole symmetrinen toisin kuin Cauchyn jännitystensori $\overline{\overline{\mathbf{T}}}$, otetaan käyttöön Piola-Kirchhoffin toinen jännitystensori [3]

$$\overline{\overline{\mathbf{\Sigma}}} = \det(\nabla\phi)(\nabla\phi)^{-1}\overline{\overline{\mathbf{T}}}(\nabla\phi)^{-T}. \quad (23)$$

Tämä voidaan kirjoittaa ensimmäisen Piola-Kirchhoffin jännitystensorin (21) avulla muotoon [3]

$$\overline{\overline{\mathbf{\Sigma}}} = (\nabla\phi)^{-1}\overline{\overline{\mathbf{T}}}_R. \quad (24)$$

Nyt voidaan referenssikonfiguraation tasapainoyhtälö (22) kirjoittaa muotoon

$$-\operatorname{div}(\nabla\phi\overline{\overline{\mathbf{\Sigma}}}) = \mathbf{f}_R. \quad (25)$$

Kun vielä huomioimme, että $\nabla\phi = \mathbf{I} + \nabla\mathbf{u}$, joka seuraa suoraan yhtälöstä (15), saadaan tasapainoyhtälöksi

$$-\operatorname{div}[(\mathbf{I} + \nabla\mathbf{u})\overline{\overline{\mathbf{\Sigma}}}] = \mathbf{f}_R \quad (26)$$

Homogeeniselle, elastiselle materiaalille pätee

$$\overline{\overline{\mathbf{\Sigma}}} = \hat{\Sigma}(\overline{\overline{\mathbf{E}}}), \quad (27)$$

missä $\hat{\Sigma}$ on toisen Piola-Kirchhoffin jännitystensorin vastefunktio. Isotrooppisille materiaaleille vastefunktio on muotoa

$$\hat{\Sigma}(\overline{\overline{\mathbf{E}}}) = \lambda(\operatorname{tr}\overline{\overline{\mathbf{E}}})\mathbf{I} + 2\mu\overline{\overline{\mathbf{E}}}, \quad (28)$$

missä

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad (29)$$

$$\mu = \frac{E}{2(1+\nu)} \quad (30)$$

ovat Lamé:n vakioita [3]. Näissä yhtälöissä E on kimmokerroin ja ν on Poisson vakio. Tämä on ns. St. Venant-Kirchhoff materiaalimalli. Kerätään vielä tarvittavat kaavat yhteen

$$\begin{cases} -\operatorname{div}[(\mathbf{I} + \nabla\mathbf{u})\overline{\overline{\mathbf{\Sigma}}}] = \mathbf{f}_R \\ \overline{\overline{\mathbf{\Sigma}}} = \lambda(\operatorname{tr}\overline{\overline{\mathbf{E}}})\mathbf{I} + 2\mu\overline{\overline{\mathbf{E}}} \\ \overline{\overline{\mathbf{E}}} = \frac{1}{2}(\nabla\mathbf{u}^T + \nabla\mathbf{u} + \nabla\mathbf{u}^T\nabla\mathbf{u}). \end{cases} \quad (31)$$

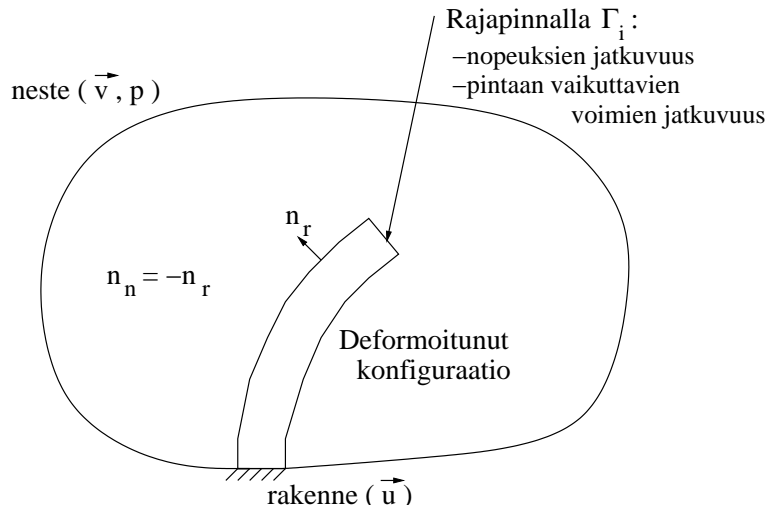
3.3 Virtaus- ja rakennemallien yhdistäminen

Virtaavan aineen ja rakenteen rajapinnan yhdistämiseen tarvitaan sekä kinemaattista- että tasapainoehtoa. Kinemaattisena ehtona on nopeuden jatkuvuus rajapinnalla ja tasapainoehtona on rajapintaan vaikuttavien voimien jatkuvuus, ks. kuva 2. Jatkuvuustilan tarkasteluissa nopeudet rajapinnalla ovat nollia. Rajapinnan jatkuvuusehdot voidaan kirjoittaa muotoon

$$\mathbf{v} = \dot{\mathbf{u}} = 0, \quad \Gamma_i \text{llä} \quad (32)$$

$$\bar{\boldsymbol{\sigma}}_r \cdot \mathbf{n}_r + \bar{\boldsymbol{\sigma}}_n \cdot \mathbf{n}_n = 0, \quad \Gamma_i \text{llä,} \quad (33)$$

missä \mathbf{n}_r on rakenteesta ulospäin suuntaava yksikkö normaali deformoituneessa konfiguraatiossa, $\dot{\mathbf{u}}$ on rakenteen siirtymisnopeus ja $\mathbf{n}_n = -\mathbf{n}_r$. Jännitystensorit $\bar{\boldsymbol{\sigma}}$ ovat Cauchy jännityksiä ts. jännityksiä deformoituneessa konfiguraatiossa [4].



Kuva 2: Virtaavan aineen ja rakenteen rajapinta [4].

Ratkaistaessa edellä mainittuja differentiaaliyhtälöitä elementtimenetelmällä tarvitaan vielä lisäksi lineaarisen elastisuusteorian mukaisia yhtälöitä elementtiverkon päivitykseen. Lineaarisen elastisuusteorian mukaan venymä voidaan kirjoittaa muotoon

$$\bar{\boldsymbol{\varepsilon}} = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u}), \quad (34)$$

missä \mathbf{u} on lineaarinen siirtymä. Jos käytetään St. Venant-Kirchhoff materiaallimallia voidaan jännitykset kirjoittaa lineaaristen venymien avulla muotoon

$$\bar{\boldsymbol{\sigma}} = \lambda \text{tr}[\bar{\boldsymbol{\varepsilon}}(\mathbf{u})]\mathbf{I} + 2\mu\bar{\boldsymbol{\varepsilon}}(\mathbf{u}). \quad (35)$$

Sijoittamalla jännitykset (35) tasapainoyhtälöön (26) ja huomioimalla lineaarisuus saadaan lineaarisen elastisuusteorian mukainen tasapainoyhtälö

$$-\text{div}\{\lambda \text{tr}[\bar{\boldsymbol{\varepsilon}}(\mathbf{u})]\mathbf{I} + 2\mu\bar{\boldsymbol{\varepsilon}}(\mathbf{u})\} = \mathbf{f}_R. \quad (36)$$

3.4 Optimointimenetelmät

Optimointimenetelmistä käsitellään tässä kahta eri menetelmää, toistettua kvadraattista optimointia (SQP) ja polytooppihakua (Downhill Simplex).

3.4.1 Toistettu kvadraattinen optimointi

Toistettu kvadraattinen optimointi (SQP) on Newtonin menetelmän yleistys rajoitteettomalle optimoinnille. Sillä voidaan kuitenkin ratkaista rajoitettuja epälineaarisia optimointiongelmia, joissa on differentioituvat kohde- ja rajoitefunktio [5].

SQP:ssä epälineaarisen tehtävän Lagrangen funktiota arvioidaan kvadraattisella mallilla. Ideana on muodostaa jono kvadraattisia tehtäviä, joiden ratkaisut lähenevät minimipistettä \mathbf{x}^* ja joiden Lagrangen kertoimet lähestyvät Lagrangen kertoimia optimipisteessä koko tehtävälle [6].

Lagrangen funktio yhtälö- ja epäyhtälörajoitteita sisältävään epälineaariseen optimointitehtävään

$$\min_{\mathbf{x}} f(\mathbf{x}), \text{ kun } \mathbf{q}(\mathbf{x}) \leq \mathbf{0} \text{ ja } \mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (37)$$

on

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \langle \mathbf{u}, \mathbf{g}(\mathbf{x}) \rangle + \langle \mathbf{v}, \mathbf{h}(\mathbf{x}) \rangle, \quad (38)$$

missä vektorit $\mathbf{u} \geq 0$ ja \mathbf{v} sisältävät Lagrangen kertoimet. Soveltamalla Taylorin lausetta Lagrangen funktioon saadaan

$$L(\mathbf{x}^k + \mathbf{p}^k, \mathbf{u}^k, \mathbf{v}^k) = \nabla L(\mathbf{x}^k, \mathbf{u}^k, \mathbf{v}^k) + (\nabla \nabla L(\mathbf{x}^k, \mathbf{u}^k, \mathbf{v}^k))\mathbf{p}^k + \dots, \quad (39)$$

missä ∇ tarkoittaa derivointia vektorin \mathbf{x} suhteen. Hakuaskel \mathbf{p}^k saadaan ratkaistua jos Taylorin sarjakehitelmästä jätetään korkeamman asteen termit huomiotta ja asetetaan $L(\mathbf{x}^k + \mathbf{p}^k, \mathbf{u}^k, \mathbf{v}^k) = 0$. Merkitsemällä Hessen matriisia $\nabla \nabla L(\mathbf{x}^k, \mathbf{u}^k, \mathbf{v}^k)$ matriisilla W_k saadaan yhtälö

$$W_k \mathbf{p}^k = -\nabla L(\mathbf{x}^k, \mathbf{u}^k, \mathbf{v}^k). \quad (40)$$

SQP:ssä ratkaistaan siten kussakin vaiheessa tehtävä

$$\min_{\mathbf{p}} \frac{1}{2} \langle \mathbf{p}^k, W_k \mathbf{p}^k \rangle + \langle \nabla f(\mathbf{x}^k), \mathbf{p}^k \rangle, \text{ kun } \mathbf{p}^k \in \mathbb{R}^n, \quad (41)$$

missä matriisi W_k on arvio Lagrangen funktion Hessen matriisille pisteessä \mathbf{x}^k . Kvadraattisen tehtävän rajoitteet saadaan linearisoimalla rajoitefunktio \mathbf{g} ja \mathbf{h} nykyisessä pisteessä \mathbf{x}^k :

$$\begin{cases} g_i(\mathbf{x}^k) + \langle \nabla g_i(\mathbf{x}^k), \mathbf{p}^k \rangle \leq 0, & i = 1, \dots, p, \\ h_j(\mathbf{x}^k) + \langle \nabla h_j(\mathbf{x}^k), \mathbf{p}^k \rangle = 0, & j = 1, \dots, q. \end{cases} \quad (42)$$

Seuraavan iteraatiopisteen löytämiseen voidaan käyttää viivahakua

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda \mathbf{p}^k, \quad \lambda \in (0, 1], \quad (43)$$

missä λ on askelpituus ja \mathbf{p}^k kertoo uuden haun suunnan [6].

SQP soveltuu ongelmiin joilla on sileät kohdefunktiot sekä pieniin ongelmiin joissa on enintään noin 100 muuttujaa [5].

3.4.2 Polytooppihaku

Polytooppihaussa (downhill simplex) muodostetaan n -ulotteinen monitahokas, ks. Haataja [6], jota kutsutaan simpleksiksi. Tätä simpleksiä sopivasti skaalamalla ja sivujen suhteen peilaamalla yritetään löytää nollakohdan sijainti.

Simpleksi sisältää $n+1$ kärkipistettä \mathbf{x}_i . Merkitään pienintä ja suurinta funktion arvoa simpleksin kärkipisteissä $f_l = \min_i f(\mathbf{x}_i)$ ja $f_h = \max_i f(\mathbf{x}_i)$ sekä lisäksi $f_i = f(\mathbf{x}_i)$. Olkoon simpleksin n parhaan kärjen painopiste $\mathbf{x}_c = \frac{1}{n} \sum_{i \neq h} \mathbf{x}_i$. Valitaan peilauskerroin $\alpha > 0$, laajennuskerroin $\gamma > 1$ ja pienennyskerroin $\beta \in [0, 1]$.

Simplex menetelmässä käytetyt peilaus- ja skaalausoperaatiot toimivat seuraavasti [6]:

1. Peilaa huonoin kärki \mathbf{x}_h vastapäätä olevan sivun suhteen:

$$\mathbf{x}_r = (1 + \alpha)\mathbf{x}_c - \alpha\mathbf{x}_h. \quad (44)$$

Jos $f_r > f_i$ kaikilla $i \neq h$, mene vaiheeseen 3. Jos taas $f_r \in [f_l, f_h]$, aseta $\mathbf{x}_h = \mathbf{x}_r$ ja toista vaihe 1 saadulle uudelle simpleksille. Muuten siirry vaiheeseen 2.

2. Jos $f_r < f_l$, suurena simpleksiä löydettyyn suuntaan:

$$\mathbf{x}_e = \gamma\mathbf{x}_r + (1 - \gamma)\mathbf{x}_c. \quad (45)$$

Jos $f_e < f_l$, aseta $\mathbf{x}_h = \mathbf{x}_e$, muuten $\mathbf{x}_h = \mathbf{x}_r$ (simpleksin suurentaminen ei kannata). Jatka vaiheesta 1.

3. Jos $f_r < f_h$, aseta $\mathbf{x}_h = \mathbf{x}_r$. Pienennä simpleksiä:

$$\mathbf{x}_s = \beta\mathbf{x}_h + (1 - \beta)\mathbf{x}_c. \quad (46)$$

Jos $f_s < \min\{f_h, f_r\}$, aseta $\mathbf{x}_h = \mathbf{x}_s$. Muussa tapauksessa aseta $\mathbf{x}_i = (\mathbf{x}_i + \mathbf{x}_l)/2$ eli puolita simpleksin sivut. Mene vaiheeseen 1.

Simplex menetelmän lopetuskriteeriksi sopii esimerkiksi [6]

$$f_h - f_l \leq \epsilon(1 + |f_l|), \quad (47)$$

missä $\epsilon > 0$.

Simplex menetelmä soveltuu tapauksiin, joissa optimoitava funktio ei ole sileä. Se on luotettava, mutta sen suppenemisnopeus saattaa olla huono [6].

4 Ratkaisu menetelmät

4.1 Käytetyt ohjelmistot

Kappaleen taipuma virtauskentässä laskettiin Elmer-ohjelmistolla. Elmer on elementtimenetelmäohjelma, jolla voidaan ratkaista virtaus-, rakenne-, sähkömagnetiikka- ja lämmönsiirto-ongelmia sekä näiden yhdistelmiä. Se koostuu kolmesta eri osasta; Elmerfront, Solver ja Elmerpost. Elmerfrontilla voidaan määritellä ongelman geometria, fysikaalinen malli, alku- ja reunaehdot. Solver ratkaisee määritellyn ongelman ja Elmerpostilla tuloksia voidaan visualisoida. Tässä erikoistyössä näistä käytettiin vain Solveria ja Elmerpostia.

Ongelman geometria ja elementtiverkko luotiin käyttäen Quickmesh ohjelmaa. Sitä käytettiin Elmerfrontin sijasta, koska ongelman geometria piti pystyä parametrisoimaan sekä ohjelmaa piti pystyä ajamaan komentoriviltä. Quickmesh on elementtiverkon generoimiseen tarkoitettu ohjelma, jolla voidaan määrittellä ongelman geometria ja verkottaa se 4, 5, 8, 9, 12 tai 16 -solmuisia suorakulmio- tai 3, 6 tai 10 -solmuisia kolmioelementtejä käyttäen.

Optimoinnissa apuna käytettiin Synapsin Pointer ohjelmistoa. Se sisältää kolme eri optimointimenetelmää: Geneettiset algoritmit, Downhill Simplexin (polytooppihaku) sekä SQP:n. Sillä voi myös suorittaa automaattisen optimoinnin, jolloin Pointer itse valitsee kuvatuunlaiselle kustannusfunktiolle ja annetulle ajalle sopivan yhdistelmän algoritmeista. Ohjelmalla on myös mahdollista kouluttaa oma, ongelmaan sopiva, optimointi proseduuri.

4.2 Palkin taipuman laskeminen

Kappaleen taipuma laskettiin elementtimenetelmällä käyttäen Elmeriä. Se ratkaisee tämän kytketyn virtaus-rakenne vuorovaikutusongelman yhtälöiden (12), (31) ja (36) avulla. Navier-Stokes yhtälöistä (12) ratkaistaan ensin virtausnopeus \mathbf{v} ja paine p . Nämä voidaan ratkaista ottamalla huomioon reunan Γ_2 jatkuvuusehto (32) sekä reunan Γ_1 reunaehdot. Yhtälön (3) avulla saadaan nyt jännitykset ratkaistua. Näiden jännitysten avulla saadaan laskettua virtauksesta reunapinnalle Γ_2 kohdistuvat voimat.

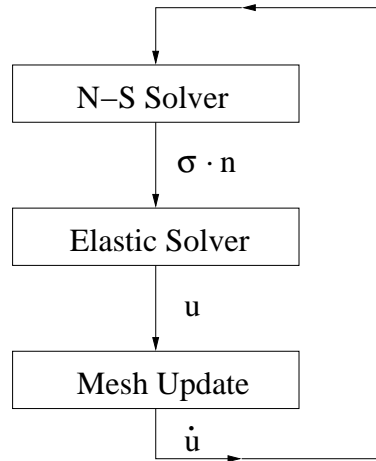
Jatkuvuusehdon (33) mukaan rajapinnalla Γ_2 virtauksesta kohdistuvat voimat ovat yhtäsuuret kappaleen reunapinnan voimien kanssa. Tästä ehdosta saadaan ratkaistua Cauchy jännitykset kappaleen reunalla. Nämä jännitykset toimivat reunaehtoina ratkaistaessa kappaleen siirtymää \mathbf{u} elastisuus yhtälöistä (31).

Kappaleen siirtymän avulla päivitetään sitten virtaavan aineen elementtiverkkoa. Tämä päivitys pitää tehdä, jotta virtaavan aineen elementtiverkkoon ei pääse syntymään päällekkäisiä elementtejä. Päivitykseen on tehty myös Fortran kielinen apufunktio, jolla estetään elementtien degeneroituminen ongelmallisen kulmapisteen läheisyydessä. Se yksinkertaisesti skaalaa elastiseksi materiaaliksi ajatellun virtaavan aineen kimmokerrointa etäisyyden funktiona valitusta pisteestä (x_0, y_0) yhtälön $1/\sqrt{(x-x_0)^2+(y-y_0)^2}$ mukaisesti. Elementtiverkon päivitykseen käytettävästä lineaarisesta elastisuusyhtälöstä (36) ratkaistaan virtaavan aineen elementtiverkon muodonmuutosnopeus $\dot{\mathbf{u}}$, joka voidaan sitten sijoittaa seuraavassa aika-askeleessa Navier-Stokes yhtälöihin (12).

Kuvassa 3 on esitetty kaavio tämän kytketyn ongelman ratkaisussa käytetystä peräkkäis-algoritmista. Yhtälöistä lineaariset elastisuus yhtälöt (36) ratkaistaan suoralla menetelmällä, joka perustuu LU-hajotelmaan. Navier-Stokes

yhtälöt(12) sekä elastisuus yhtälöt(31) ratkaistaan iteratiivisesti. Iterointiproseduuri koostuu Järvinen, Lyly, Ruokolainen ja Råback [2] mukaan aikaintegroinnista ja kolmesta sisäkkäisestä iterointisilmukasta (heikko kytkentä). Aluksi ongelma diskretoidaan implisiittisesti ajan suhteen. Jokaisella aika-askeleella suoritetaan seuraavat toimenpiteet:

- Jaetaan ongelma pääräkkäisiin kytkemättämiin aliongelmiksi kuten kuvassa 3 on näytetty.
- Ratkaistaan epälineaariset aliongelmat (Navier-Stokes- ja elastisuusyhtälöt) iteratiivisesti peräkkäisillä linearisoinneilla käyttäen Picardin ja Newtonin iteraatioita. Lineaarinen elastisuusyhtälö ratkaistaan suoraan ilman iterointia.
- Jokaisella linearisointi kerralla: Ratkaistaan linearisoitu elastisuusyhtälö suoraan LU-hajotelmalla ja Navier-Stokes iteratiivisesti käyttäen stabiloitua bikonjugaattigradienttimenetelmää.



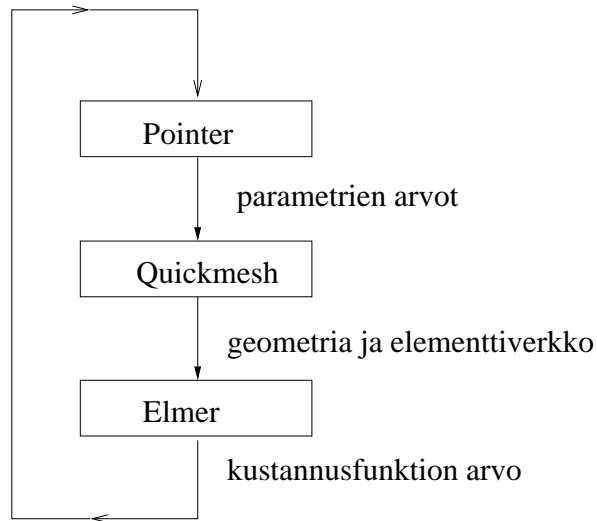
Kuva 3: Kaavio taipuman ratkaisuperiaatteesta [2].

Kunkin iterointisilmukan suoritus päättyy, kun ratkaisun tarkkuus saavuttaa kyseiselle silmukalle ennalta määrätyn konvergenssitoleranssin. Uloimman silmukan suorituksen päättää jatkuvuustilan (steady state) konvergenssitoleranssi, T_{SS} . Keskimmäisen silmukan suoritus päättyy kun virheestä tulee pienempi kuin epälineaarisen systeemin konvergenssitoleranssi, T_{NL} . Lineaarisen systeemin konvergenssitoleranssi T_L taas päättää sisimmän iteraatiosilmukan suorituksen. Siltä varalta, että laskettava yhtälö ei konvergoi, on myös suoritettavien iteraatioiden maksimi lukumäärä rajattu. Näin vältetään päättymättömiltä silmukoilta.

4.3 Optimoinnin suorittaminen

Optimoinnin suoritusta on havainnollistettu kuvassa 4. Pointer suorittaa itse optimoinnin määrättyjen parametrien ja kustannusfunktion avulla. Se pitää kirjaa

parhaista optimointi parametrien arvoista sekä laskee uusien kokeiltavien parametrien arvot, jotka se sitten antaa Quickmeshille. Quickmeshin avulla kappaleen optimoitava reuna Γ_2 on parametrisoitu. Se luo Pointerin antamilla uusilla parametrien arvoilla kappaleelle geometrian ja koko ongelman geometrialle uuden elementtiverkon. Tällä uudella geometrialla ja elementtiverkolla Elmer sitten laskee kappaleen taipuman ja sen avulla uuden kustannusfunktion arvon, jonka se sitten palauttaa Pointerille. Tätä toistetaan kunnes jokin Pointerin sisältämistä optimoinnin lopetuskriteereistä toteutuu. Parhaiten kontrolloitavissa oleva lopetuskriteeri Pointerissa on aika, joka optimointiin saa kulua.



Kuva 4: Kaavio ohjelmistojen yhdistämistavasta.

Optimoinnin automatisoimiseksi kaikki käytetyt ohjelmistot (Elmer, Quickmesh ja Pointer) piti yhdistää. Ohjelmistojen yhdistäminen tapahtui Pointerin avulla. Se saadaan ajamaan sekä Elmeriä että Quickmeshiä komentoriviltä. Sen avulla pystytään myös vaihtamaan tietoja optimoitavien parametrien sekä kustannusfunktion arvoista eri ohjelmien välillä. Ainoana käyttäjän käyttämänä ohjelmassa itse optimoinnissa toimi siis vain Pointer.

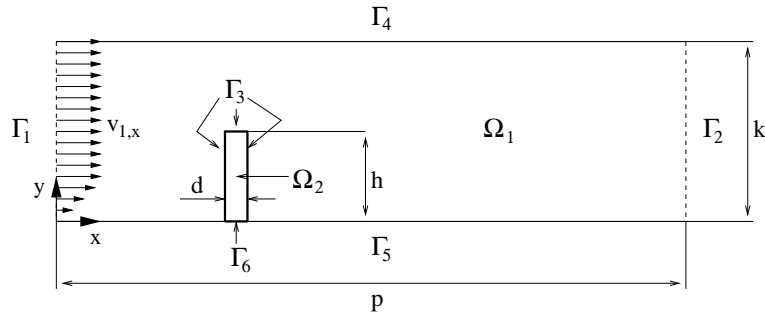
5 Esimerkkiongelman geometria

5.1 Geometria ja alkuperäinen taipuma

Esimerkkiongelman tutkittiin kuvan 5 mukaista 2D-geometriaa. Siinä homogeeninen, elastinen palkki Ω_2 on virtauskentässä Ω_1 . Palkki on jäykästi kiinnitetty alustaan päädyistä Γ_6 ja tarkastelualueen sisääntuloreunalla Γ_1 virtaavan aineen nopeus x-suuntaan kasvaa jatkuvasti nolasta maksimiarvoonsa $v_{1,x}$. Ennen maksimiarvonsa saavuttamista virtausnopeus määritellään yhtälön

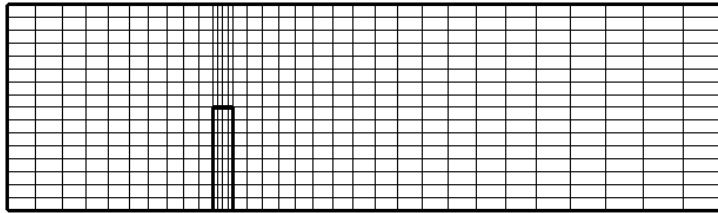
$$a(-y^2 + by)/c$$

mukaan, missä a , b ja c ovat vakioita. Sisääntulovirtauksen nopeuden määrittely on toteutettu Fortran-kielisenä funktiona InFlow.



Kuva 5: Esimerkkiongelman geometria.

Palkin alkuperäistä taipumaa ratkaistaessa elementtiverkkona oli kuvan 6 mukainen verkko. Elementteinä käytettiin kahdeksansolmuisia suorakaide-elementtejä.

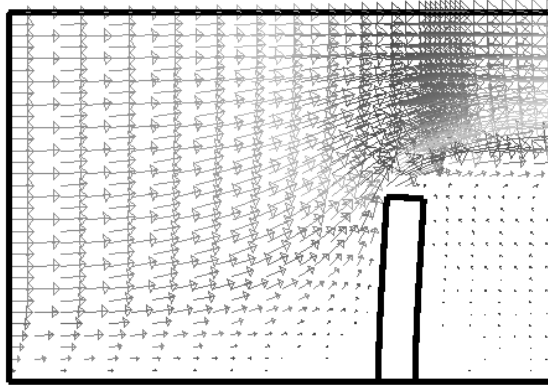


Kuva 6: Elementtiverkko. Elementit ovat kahdeksansolmuisia suorakaide-elementtejä.

Alkuperäisellä geometrialla palkin taipumaksi w saatiin 2.36 mm. Kuvassa 7 on havainnollistettu palkin taipumaa. Siitä näkee myös hyvin virtauksen sisääntuloprofilin muodon.

5.2 Kustannusfunktio

Tehtävänä oli siis minimoida palkin taipuma w . Jotta palkin pinta-alaan liittyvä rajoite tulee huomioiduksi, etsitään kustannusfunktioksi funktiota, joka kuvaa



Kuva 7: Palkin taipuma alkuperäisellä muodolla sekä virtauksen tuloprofiili.

palkin taipumaa pitäen kuitenkin samalla palkin pinta-alan A vakiona. Pinta-alan kasvaminen pitää siis kasvattaa kustannusfunktion arvoa sopivassa suhteessa taipuman muutokseen. Vaadimme, että palkin korkeus h ja pinta-ala pysyvät vakioina, mutta leveys d saa muuttua koko korkeuden h pituudella.

Tiedämme, että palkin ala on suoraan verrannollinen sen leveyteen, $A \propto d$, sekä palkin taipuma kääntäen verrannollinen leveyden kolmanteen potenssiin, $w \propto \frac{1}{d^3}$. Tämän perusteella oletamme, että $wA^3 \approx \text{vakio}$. Tämä pätee kuitenkin vain suorakulmion muotoisille palkeille. Koska palkin geometria muuttuu optimoinnin aikana, on muutkin geometriat pystyttävä ottamaan huomioon. Tätä varten otetaan käyttöön geometriakerroin g , joka kuvaa palkin leveyden d muuttumista koko palkin korkeuden matkalta. Kaikille geometrioille pätee

$$gwA^3 \approx g_0w_0A_0^3, \quad (48)$$

missä alaindeksi 0 viittaa alkuperäiseen geometriaan. Koska alkuperäinen palkki on suorakulmion muotoinen on $g_0 = 1$ ja geometriakerroimeksi saadaan

$$g = \frac{w_0A_0^3}{wA^3}. \quad (49)$$

Johdetaan nyt kustannusfunktio ensin suorakulmion muotoisille palkeille ja laajennetaan se sitten geometriakerroimen avulla pätemään myös muillekin geometriolle. Muistetaan, että suorakulmion muotoisilla palkeilla pätee $wA^3 \approx \text{vakio}$.

Olkoon alalla A laskettu taipuma w . Arvioidaan vastaavaa taipumaa w_0 , kun ala on A_0 . Taipumaa yliarvoidaan aina hieman, jolloin saadaan

$$w_0 = w + (w_0 - w) \quad (50)$$

$$= w + \left(\frac{wA^3}{A_0^3} - w\right) \quad (51)$$

$$= w + w\left(\frac{A^3}{A_0^3} - 1\right). \quad (52)$$

Koska alkuperäisen alan A_0 alituksesta ei tarvitse sakottaa, voidaan yhtälö (52) kirjoittaa muotoon

$$w_0 = w[1 + \max(0, \frac{A^3}{A_0^3} - 1)]. \quad (53)$$

Kun yhtälössä (53) otetaan nyt muutkin geometriat huomioon, geometriakerroimen g avulla, saadaan

$$w_0 = w[1 + g \max(0, \frac{A^3}{A_0^3} - 1)]. \quad (54)$$

Sijoittamalla geometriakerroin yhtälöstä (49) yhtälöön (54) saadaan kustannusfunktiksi

$$f = w_0 = w + w_0 \max(0, 1 - \frac{A_0^3}{A^3}) \quad (55)$$

Tämäkään yhtälö ei vielä mitenkään suosi alkuperäistä alaa A_0 . Tarvitaan siis sakkokerroin c , jolla sakotetaan sopivasti alkuperäisen alan ylityksestä. Tällöin kustannusfunktio saadaan lopulliseen muotoonsa

$$f = w + c w_0 \max(0, 1 - \frac{A_0^3}{A^3}). \quad (56)$$

Kustannusfunktioa (56) katsomalla voidaan todeta, että funktio saa aina arvokseen kyseisellä geometrialla lasketun taipuman arvon sekä, pinta-alan ollessa alkuperäistä alaa suurempi, myös sakkotermiksi osan alkuperäisellä geometrialta lasketusta taipumasta. Saadussa kustannusfunktiossa on siis toivotut ominaisuudet. Käytäntö osoitti kyseisen funktion hyvin toimivaksi sakkokertoimen saadessa arvon $c = 1.5$. Alkuperäisellä geometrialla lasketun taipuman arvoksi saatiin Elmerillä laskettuna $w_0 = 2.36$ mm.

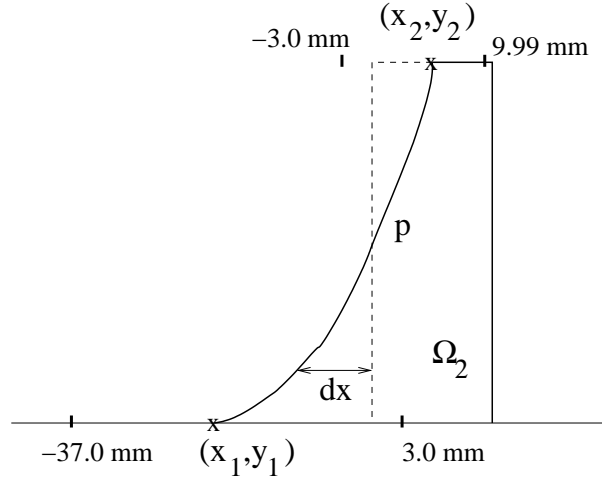
Kustannusfunktion arvon laskenta liitettiin Elmeriin. Se suorittaa aina taipuman laskemisen päätteeksi Fortran-kielisen funktion Costfunction, jossa lasketun taipuman avulla lasketaan kustannusfunktiolle arvo yhtälön (56) mukaan.

5.3 Optimointi

Optimoitavana muotona käytettiin palkin virtauksen tulopuolella olevaa kylkeä. Se kuvattiin ensin paloittaisen potenssilain ja sitten sinisarjan avulla.

Paloittaista potenssilakia käytettäessä suunnitteluparametrejä oli kolme kappaletta x_1 , x_2 ja p . Parametreistä x_1 kuvaa palkin optimoitavan kyljen alimman pisteen siirtymää x-akselin suunnassa, x_2 kuvaa palkin optimoitavan kyljen ylimmän pisteen siirtymää x-akselin suunnassa ja p on potenssi. Näiden suunnitteluparametrien avulla interpoloidaan pisteet dx pisteiden y_1 ja y_2 välillä yhtälön $a(y - y_1)^p$ mukaan, missä $a = \frac{x_2 - x_1}{y_2^p - y_1^p}$ (ks. kuva 8). Optimointia varten siirtymä x_1 rajattiin välille -37.0 mm - 3.0 mm, siirtymä x_2 välille -3.0 mm - 9.99 mm ja potenssi p välille 0.4 - 2.5. Suunnitteluparametreille annettuihin rajoihin ei optimoinnissa törmätty, joten tehtävä oli kyseisellä kustannusfunktiolla hyvin asetettu.

Käytettäessä palkin kyljen kuvaamiseen sinisarjaa jouduttiin optimointiin liittyvistä käytännön syistä johtuen tekemään ensin suora, jonka määritteli kaksi suunnitteluparametriä x_1 ja x_2 . Tällä suoralla mahdollistettiin kyljen kallistuminen. Suoran määräämien päätepisteiden väliin kuvattiin sitten kyljen muoto sinisarjan neljää ensimmäistä termiä käyttäen. Käytetty sinisarja voidaan



Kuva 8: Palkin kyljen muodon optimoinnin periaate paloittaisen potenssilain avulla.

esittää muodossa $\sum_{n=1}^4 dx_n \sin(\frac{n\pi y}{h})$, missä h on määritellyn suoran pituus ja dx_n :t ovat suunnitteluparametrejä, joilla määritellään kunkin siniaallon amplitudi. Suunnitteluparametrit x_1 ja x_2 kuvaavat samoja pisteitä kuin kuvassa 8 esitetyt pisteet. Ne ovat myös rajattu optimointia varten samoille väleille kuin vastaavat pisteet potenssilakia käytettäessä. Siniaaltojen amplitudeista dx_1 on rajattu välille $-20.0 \text{ mm} - 2.0 \text{ mm}$. Muut amplitudit on rajattu välille $-20.0 \text{ mm} - 3.0 \text{ mm}$. Tälläkään parametrisoinnilla ei annettuihin rajoihin optimoinnissa törmätty, joten tehtävä oli kyseisellä kustannusfunktiolla hyvin asetettu.

Optimointiajoja Pointerilla suoritettiin kuusi kappaletta molemmilla kyljen parametrisointi tavoilla eli yhteensä 12 kpl. Optimoinnit suoritettiin kolmella eri Pointerin optimointitavalla. Koska kustannusfunktio on jatkuva ja sileä ei optimoinnissa käytetty Pointerista löytyvää geneettistä algoritmiä vaan optimoinnit suoritettiin SQP:llä (toistettu kvadraattinen optimointi), polytooppihaulla sekä Pointerista löytyvällä automaattisella optimoinnilla.

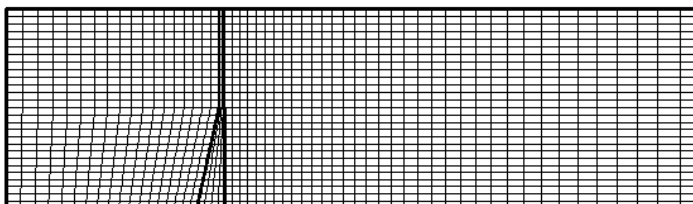
Optimoidessa potenssilaila parametrisoitua palkin kylkeä käytettiin aloitusparametrien arvoina alkuperäistä palkin geometriaa vastaavia arvoja $x_1 = 0.0 \text{ mm}$, $x_2 = 0.0 \text{ mm}$ ja $p = 1.0$. Kaksi ensimmäistä optimointia suoritettiin käyttäen polytooppihakua (simplex). Ensimmäisen simplex-haun askelpituutena käytettiin arvoa 0.1. Askelpituudet ovat suhteellisia liittyen annettuihin rajoihin. Toinen simplex-haku aloitettiin ensimmäisen haun optimiparametrien arvoista ja se suoritettiin askelpituudella 0.01. Seuraavat kaksi optimointia suoritettiin SQP:llä. Samaan tapaan kuin simplex-hauilla ensimmäisen SQP:llä suoritettun optimoinnin askelpituus oli 0.1 ja toisen 0.01. Toisen SQP:llä suoritettun optimoinnin aloitusparametreinä käytettiin ensimmäisen SQP-haun optimiparametreja. Viimeiset kaksi optimointia suoritettiin käyttämällä Pointerin automaattista optimointia. Molemmissa optimoinneissa kustannusfunktion kerrottiin olevan jatkuva ja sileä. Toisen automaattisen optimoinnin aloitusparametreinä käytettiin ensimmäisellä Pointerin automaattisesti suorittamalla optimoinnilla saatuja optimiparametrien arvoja. Tiedot potenssilaila kuvatun kyljen muodon optimoinnissa käytetyistä menetelmistä on kerätty taulukkoon 1.

Sinisarjan käyttö mahdollisti monimutkaisemmat palkin kyljen muodot. Täs-

Taulukko 1: Käytetyt optimointi menetelmät potenssilaila kuvatun kyljen muodon optimoinnissa.

	Simplex	Simplex	SQP	SQP
Askelpituus	0.1	0.01	0.1	0.01
Iteraatioita	35	25	10	10
Aloituseram.	alkuper.	optimi	alkuper.	optimi

tä johtuen käytettäessä sinisarjaa kyljen muodon parametrisointiin havaittiin, että taipumaa ei enää pystytä ratkaisemaan alkuperäisellä elementtiverkolla vaan verkkoa piti tihentää. Tämä kuitenkin lisäsi yhden taipumaratkaisun laskemiseen kuluvaa aikaa niin paljon, että ratkaisu päätettiin laskea suoraan yhdellä iteraatiolla. Tämä merkitsee, että muuttuneen geometrian vaikutusta virtaukseen ei huomioida. Optimoinnin aloitusparametreinä käytettiin virtauksen suuntaan vinoa suoraa kuvaavan geometrian parametrien arvoja $x_1 = -4.0$ mm ja $x_2 = 7.0$ mm. Tällöin siniamplitudit $dx_n = 0$ mm, $n = 1, \dots, 4$. Kuvassa 9 on esitetty tihennetty verkko ja optimoinnin aloitusgeometria. Muuten optimointiajot suoritettiin samaan tapaan kuin potenssilakia käytettäessä. Kaksi ensimmäistä optimointia suoritettiin simplex-haulla, kaksi seuraavaa SQP:llä ja kaksi viimeistä Pointerin automaattisella optimoijalla. Jokaisen menetelmän toinen optimointikerta aloitettiin edellisellä kerralla saaduista optimiparametrien arvoista. Taulukkoon 2 on kasattu tiedot sinisarjalla kuvatun kyljen muodon optimoinnissa käytetyistä menetelmistä.



Kuva 9: Tihennetty elementtiverkko ja optimoinnin aloitusgeometria käytettäessä sinisarjaa kuvaamaan optimoitavan kyljen muotoa.

Taulukko 2: Käytetyt optimointi menetelmät sinisarjalla kuvatun kyljen muodon optimoinnissa.

	Simplex	Simplex	SQP	SQP
Askelpituus	0.1	0.01	0.1	0.01
Iteraatioita	20	20	3	3
Aloituseram.	alkuper.	optimi	alkuper.	optimi

6 Esimerkkiongelman numeerinen ratkaiseminen

6.1 Mallin mitat, reunaehdot ja materiaaliominaisuudet

Esimerkkiongelman palkin korkeus $h = 0.05$ m ja palkin alkuperäinen leveys $d = 0.01$ m. Tarkastelualueen pituus $p = 0.35$ m ja korkeus $k = 0.1$ m. Elementtejä alkuperäisessä elementtiverkossa oli 528 kpl ja suoritettun tihennyksen jälkeen 1539 kpl.

Reunaehtoina esimerkkiongelman tarkastelualueen sisääntuloreunalla Γ_1 virtaavan aineen nopeuden maksimi-arvo $v_{1,x} = 1.0$ m/s, jonka se saavuttaa pisteessä $y = 0.025$ m. Tämän jälkeen virtausnopeus pysyy vakiona. Sisääntulovirtauksen nopeuden määrittelevässä funktiossa InFlow käytettyjen vakioden a , b ja c arvot olivat 1.0 1/s, 0.05 m ja 0.000625 m. Sisään- ja ulostuloreunoilla Γ_1 ja Γ_2 sekä reunalla Γ_4 on y-suuntainen nopeus $v_y = 0$ m/s. Virtaavan aineen nopeus on nolla palkin kyljillä Γ_3 sekä tarkastelu alueen reunalla Γ_5 .

Esimerkkiongelman virtaava aine on newtonilainen ja kokoonpuristumaton. Sen tiheys $\rho_f = 900$ kg/m³ ja dynaaminen viskositeetti $\mu = 0.4$ kg/ms. Sekä tiheys että viskositeetti pysyvät vakioina. Kimmokertoimen määrittelevässä funktiossa Young käytetty referenssipiste (x_0, y_0) oli $(0.105, 0.049)$. Palkin materiaaliominaisuudet ovat tiheys $\rho_b = 2700$ kg/m³, Poisson vakio $\nu = 0.3$ ja kimmokerroin $E = 1.0 \cdot 10^7$ N/m².

Eri iterointisilmukat päättävät konvergenssitoleranssit ovat $T_{SS} = 10^{-4}$, $T_{NL} = 10^{-5}$ ja $T_L = 10^{-5}$. Ne on määritelty Solver input tiedostossa (.sif), joka on liitteenä 1. Tässä tiedostossa on myös määritelty eri yhtälöiden ratkaisuissa käytetyt menetelmät.

6.2 Tulokset

Taulukkoon 3 on kerätty optimoinnin tuloksia eri optimointimenetelmillä, kun palkin kyljen muoto oli parametrisoitu potenssilaila. Aloituspärametreinä käytettiin alkuperäisen palkin geometriaa vastaavia parametrien arvoja $x_1 = 0.0$ mm, $x_2 = 0.0$ mm ja $p = 1.0$.

Taulukko 3: Tulokset eri optimointi algoritmeilla, kun palkin kyljen muoto esitetään potenssilain avulla. Aloituspärametrien arvoina alkuperäisen palkin muodon parametrien arvot.

	Simplex	SQP	Automaattinen
x_1 [mm]	-5.517	-9.092	-7.160
x_2 [mm]	9.433	8.911	7.136
p	1.709	0.976	0.997
Kustannusfunktio f [mm]	0.862	0.920	0.958
Pinta-ala A [cm ²]	4.999	4.991	4.999
Taipuma w [mm]	0.862	0.920	0.958

Taulukosta 3 nähdään, että pienin taipuma löytyi simplex-menetelmällä. Tuloksista voidaan myös havaita, että eri menetelmillä saatiin erimuotoisia optimaalisen palkin muotoja.

Taulukkoon 4 on kerätty optimoinnin tuloksia eri optimointimenetelmillä, kun palkin kyljen muoto oli parametrisoitu sinisarjan avulla. Aloituspärametreinä käytettiin virtauksen suuntaan vinoa suoraa kuvaavan geometrian vastaa-

via parametrien arvoja $x_1 = -4.0$ mm ja $x_2 = 7.0$ mm. Sinitermien amplitudit dx_n olivat kaikki nolliä.

Taulukko 4: Tulokset eri optimointi algoritmeilla, kun palkin kyljen muoto esitetään sinisarjan avulla. Aloituspäätämien arvoina virtauksen suuntaan kallellaan olevan suoran parametrien arvot.

	Simplex	SQP	Automaattinen
x_1 [mm]	-6.734	-5.866	-5.814
x_2 [mm]	7.711	6.681	6.874
dx_1 [mm]	-0.592	-0.147	-0.704
dx_2 [mm]	0.136	0.538	-0.000
dx_3 [mm]	-0.558	-1.368	-0.315
dx_4 [mm]	1.477	0.000	0.000
Kustannusfunktio f [mm]	0.911	1.012	0.958
Pinta-ala A [cm ²]	5.004	4.988	4.992
Taipuma w [mm]	0.904	1.012	0.958

Taulukosta 4 havaitaan, että paras tulos käytettäessä sinisarjaa on myös saatu simplex-menetelmällä. Koska optimoidun palkin muotoja on vaikea hahmottaa parametrien arvoista on kuvia eri optimointi menetelmillä saaduista palkin muodoista kerätty liitteeseen 2.

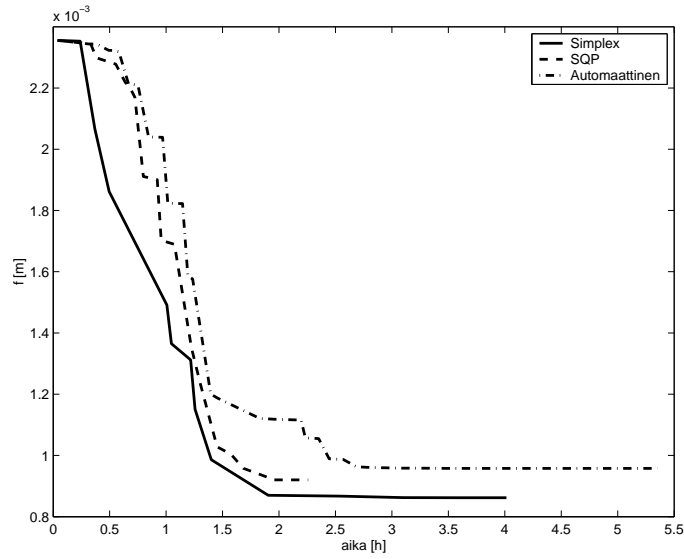
Tuloksista voidaan todeta kustannusfunktion toimivuus. Jokaisella menetelmällä palkin pinta-ala on pysynyt lähellä alkuperäisen palkin pinta-alaa, joka oli 5.000 cm². Pienin taipuma löytyi simplex-menetelmällä, mutta kaikilla muillakin menetelmillä saavutettiin huomattava parannus alkuperäisen palkin taipumaan, joka oli 2.36 mm.

Koska eri parametrisoinneilla käytettiin eri elementtiverkkoa ja ratkaisutapa muutenkin muutettiin, ei saatuja tuloksia voida suoraan verrata toisiinsa. Tarkistuksen vuoksi ratkaistiin potenssilailia parametrisoidut tapaukset myös tiheimmällä verkolla. Kun palkin taipuma ratkaistiin löydettyjä optimimuotoja käyttäen, taipumaksi saatiin simplexillä 0.849 mm, SQP:llä 0.911 mm ja automaattisella optimoinnilla 0.946 mm. Nämä kaikki ovat hieman pienempiä kuin alkuperäisellä ratkaisutavalla saadut taipumat. Vertailuissa siis potenssilailia parametrisoitujen tapausten taipumat ovat vähän liian suuria.

Vertaamalla eri parametrisoinnilla saatuja tuloksia taulukoista 3 ja 4 voimme havaita, että Pointerin automaattinen optimoija löysi yhtä hyvän tuloksen sekä potenssilakia että sinisarjaa parametrisoinnissa käytettäessä. Huomioimalla kuitenkin vertailussa edellä mainittu eri ratkaisutavoista johtuva ero voidaan todeta, että potenssilakia käytettäessä on löydetty kaikilla optimointimenetelmillä parempia tuloksia.

Eri optimointi menetelmien tehokkuutta voidaan tarkastella kuvista 10 ja 11. Tehokkuus tarkastelut suoritettiin käyttäen potenssilailia parametrisoidun kyljen optimointi historiaa. Kyseinen optimointi oli nopeampi tehdä pienemmän suunnitteluparametrien määrän takia.

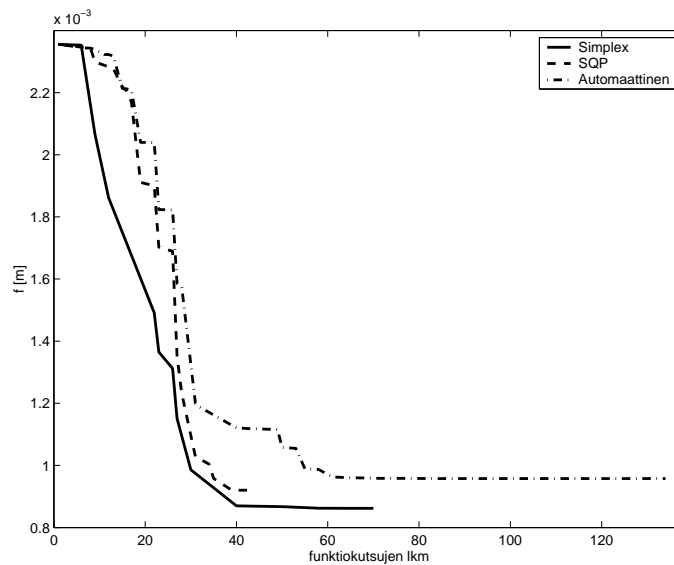
Kuvassa 10 on kustannusfunktion arvo ajan funktiona. Siitä näemme, että simplex-menetelmällä päästiin myös nopeiten parhaaseen tulokseen. Kuitenkin on syytä huomata, että SQP:llä optimointi oli kokonaisuudessa nopein ja sillä päästiin kohtuulliseen tulokseenkin. Pointerin oma automaattinen optimoija ei käyttämästään ajasta huolimatta pystynyt löytämään simplex-menetelmällä



Kuva 10: Kustannusfunktion arvo optimointiin käytetyn ajan funktiona.

saatuun optimiin verrattavaa tulosta.

Kuvassa 11 on kustannusfunktion arvo funktiokutsujen funktiona. Siitä voimme havaita, että optimiarvot löytyvät noin 40:llä funktiokutsulla sekä simplex- että SQP-menetelmillä. Tässä vaiheessa Pointerin automaattinen optimoija on vielä pahasti jäljessä sekä simplex- että SQP-menetelmän tulosta. Funktiokutsujen määrä on suoraan verrannollinen käytettyyn aikaan, joka näkyy hyvin vertaamalla kuvia 10 ja 11.



Kuva 11: Kustannusfunktion arvo optimoinnissa käytettyjen funktiokutsujen määrän funktiona.

Jotta aikaisemmin taulukoissa 3 ja 4 esitettyjä tuloksia saataisiin vielä tarkennettua, jatkettiin optimointia kullakin menetelmällä saatujen optimiparametrien arvoista, pienentäen samalla askelpituutta. Pointerin automaattisessa optimoinnissa askelpituuteen ei voitu vaikuttaa, mutta optimitulosta yritettiin silti parantaa. Esimerkiksi SQP:llä käytettäessä parametrisointiin potenssilakia aloitusparametreinä käytettiin arvoja $x_1 = -9.092$ mm, $x_2 = 8.911$ mm ja $p = 0.976$, jotka löytyvät suoraan taulukosta 3. Näiden tarkennettujen optimointiajojen tulokset on kerätty taulukkoihin 5 ja 6.

Taulukko 5: Tarkennetut tulokset, kun parametrisointi on tehty potenssilaila. Kunkin eri menetelmän aloitusparametrien arvoina oli edellisen samalla menetelmällä tehdyn optimoinnin optimiarvot.

	Simplex	SQP	Automaattinen
x_1 [mm]	-5.513	-9.124	-7.160
x_2 [mm]	9.433	8.920	7.136
p	1.712	0.977	0.997
Kustannusfunktio f [mm]	0.862	0.916	0.958
Pinta-ala A [cm ²]	5.000	4.999	4.999
Taipuma w [mm]	0.862	0.916	0.958

Taulukko 6: Tarkennetut tulokset, kun parametrisointi on tehty sinisarjan avulla. Kunkin eri menetelmän aloitusparametrien arvoina oli edellisen samalla menetelmällä tehdyn optimoinnin optimiarvot.

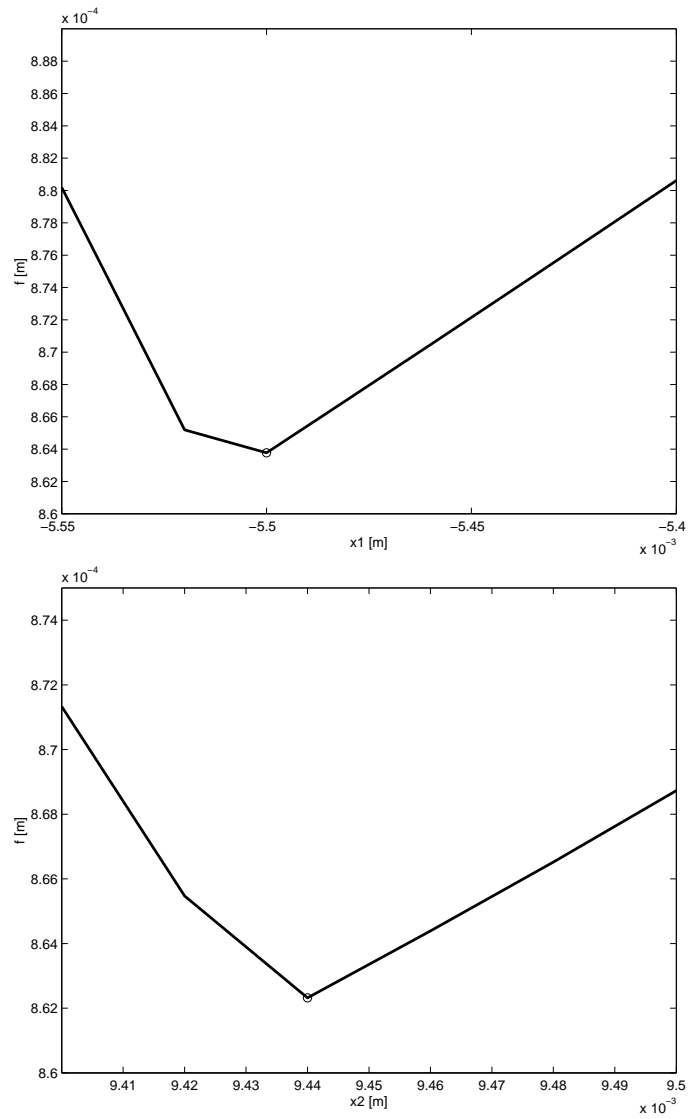
	Simplex	SQP	Automaattinen
x_1 [mm]	-6.733	-5.897	-5.834
x_2 [mm]	7.839	6.679	6.872
dx_1 [mm]	-0.664	-0.160	-0.711
dx_2 [mm]	0.311	0.536	-0.000
dx_3 [mm]	-0.624	-1.346	-0.318
dx_4 [mm]	1.517	0.000	0.000
Kustannusfunktio f [mm]	0.899	1.004	0.953
Pinta-ala A [cm ²]	5.001	4.998	5.000
Taipuma w [mm]	0.898	1.004	0.953

Taulukoita 3 ja 5 vertailemalla nähdään, että potenssilaila parametrisoidussa tapauksessa nämä jatkettut optimoinnit paransivat tarkkuuden rajoissa vain SQP:llä saatuja optimiparametrien arvoja. Niissäkään ei kovin merkittävää parannusta tapahtunut. Muissa menetelmissä tarkennukset eivät enää muuttaneet aikaisemmin saatua taipumaa.

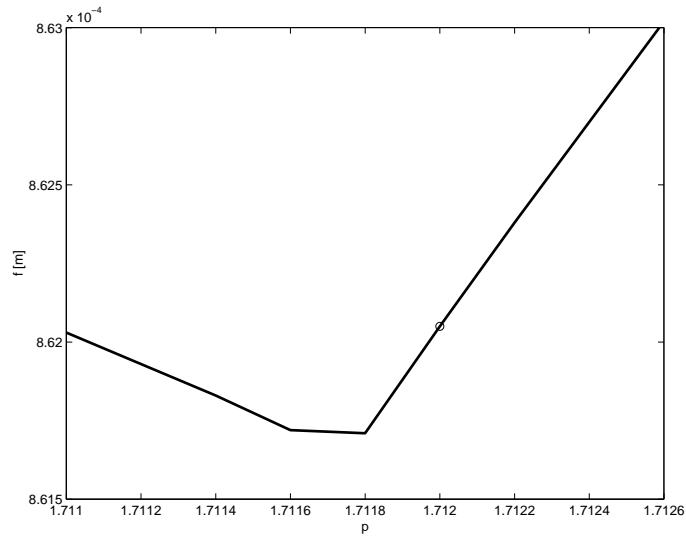
Taulukoista 4 ja 6 taas havaitaan, että sinisarjan avulla parametrisoidussa tapauksessa kaikilla optimointi menetelmillä tarkennuksella saavutettiin hieman parannusta. Kuitenkaan mitään merkittäviä muutoksia taipumiin ei tullut.

Parhaaksi tulokseksi jäi näin ollen potenssilaila parametrisoidun kyljen simplex-menetelmällä löydetty optimimuoto, jolloin palkin taipuma oli 0.862 mm. Tämän saadun optimimuodon optimiparametrien ympäristöjä tutkimalla vielä tarkistettiin, että kyseisellä optimointi menetelmällä oltiin löydetty ainakin lokaali minimi. Kuvissa 12 ja 13 olevista kuvaajista voidaan karkeasti havaita, että jokaisella parametrillä kustannusfunktio f saavuttaa miniminsä lähellä

saatua optimiparametrin arvoa. Näin ollen voitiin varmistua ainakin lokaalin minimipisteen löytymisestä.

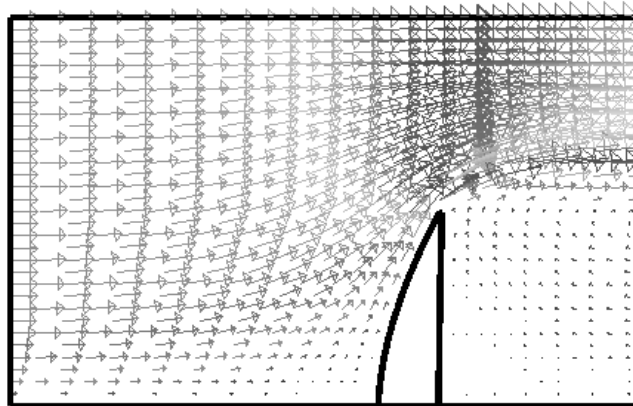


Kuva 12: Optimiparametrien $x_1 = -5.513$ mm ja $x_2 = 9.433$ mm tarkistus.



Kuva 13: Optimiparametrin $p = 1.712$ tarkistus.

Kuvassa 14 on visualisoitu vielä saatua optimaalisen palkin muotoa sekä taipumaa. Vertaamalla tätä kuvaa alkuperäistä palkin muotoa ja taipumaa esittävään kuvaan 7 voidaan selkeästi havaita optimoidulla muodolla saavutettava taipuman pieneneminen. Liitteessä 2 on kuvia muilla optimointi menetelmillä saaduista palkin muodoista.



Kuva 14: Optimoidun palkin taipuma ja muoto.

7 Pohdinnat

Tässä erikoistyössä tutkittiin kappaleen optimimuodon etsimistä virtauskentässä siten, että kappaleen maksimi taipuma olisi mahdollisimman pieni. Kappaleen taipuman laskeminen suoritettiin Elmer-ohjelmistolla elementtimenetelmää käyttäen ja optimointi automatisoitiin Pointer-ohjelmistolla, jonka kahta optimointialgoritmiä hyödynnettiin optimimuodon etsinnässä.

Esimerkkiongelmassa optimimuodon etsintää rajoitti yhden optimimuodon etsimiseen käytettävän ajan rajoittuneisuus. Pitemmillä optimointiajoilla olisi voitu löytää vielä parempia tuloksia. Pointer-ohjelmistosta johtuvien käytännön ongelmien vuoksi eräajojen ajaminen oli kuitenkin vaikeaa, jolloin numeerisissa laskelmissa tyydyttiin hiukan lyhyemmillä optimointiajoilla saatuihin tuloksiin.

Sinisarjan termejä lisäämällä olisi esimerkkiongelmahan myös voitu löytää vielä parempia tuloksia. Termien lisääminen olisi mahdollistanut suuremman muotoavaruuden, josta optimimuotoja etsittiin. Tämä olisi kuitenkin lisännyt optimointiin kuluva aikaa, joten tyydyttiin sinisarjan neljään ensimmäiseen termiin. Käytetyt parametrisoinnit mahdollistivat jo kuitenkin monien intuitiivisesti hyvien muotojen käytön.

Numeerissa laskuissa saatu paras optimimuoto on intuitiivisesti järkevä. Siinä palkki ohjaa virtausta siten, että palkin alapää, joka on jäykin, muuttaa virtaussuuntaa palkin pinnan suuntaiseksi. Tällöin palkin yläpää voi kaventua taipuman kuitenkaan kasvamatta. Palkin kyljen optimimuoto pyrkii olemaan siis mahdollisimman virtaviivainen.

Jatkossa voisi vielä tutkia pysyykö palkin kyljen optimimuoto likimain samana, kun taipumaa lisätään pienentämällä palkin jäykkyyttä. Myös muita palkin kyljen muotoja mahdollistavien sovitusten käyttöä kannattaisi kokeilla. Tällöin saataisiin suurempi skaala kokeiltavia muotoja mukaan optimointiin.

Viitteet

- [1] Hämäläinen Jari ja Järvinen Jari. 1994. *Elementtimenetelmä virtauslaskennassa*. CSC-Tieteellinen laskenta Oy. 212 s. ISBN 952-9821-07-7.
- [2] Esko Järvinen, Mikko Lyly, Juha Ruokolainen, Peter Råback. 2001. *Three-dimensional fluid-structure interaction modelling of blood flow in elastic arteries*. ECCOMAS Computational Fluid Dynamics Conference 2001, Swansea, Wales, UK, 4-7 September 2001.
- [3] Dietrich Braess. 2001. *Finite elements*. 2.p. Cambridge, United Kingdom. Cambridge University Press. 352 s. ISBN 0-521-01195-7
- [4] Lund Erik, Møller Henkik ja Jakobsen Lars Aæs. 2001. *Shape Design Optimization of Steady Fluid-Structure Interaction Problems With Large Displacements*. American Institute of Aeronautics and Astronautics, Inc. 11 s.
- [5] Synaps, Inc. 2000. *Synaps Pointer User's Guide*. 119 s.
- [6] Haataja Juha. 1995. *Optimointitehtävien ratkaiseminen*. 2. p. CSC - Tieteellinen laskenta Oy. 232 s. ISBN 952-9821-02-6.

Liitteet

1. Solver input file (.sif).
2. Eri optimointi algoritmeilla saatuja palkin muotoja.

```

! Määritellään yleisiä tietoja.
Header
  Mesh DB ".fsi"
  Bodies                2
  Equations              2
  Solvers                4
  Materials              2
  Boundaries             6
  Boundary Conditions 6
End
! Vakioiden määrittelyt.
Constants
  Gravity
  size 4
  Real 0 -1 0 9.81

  Stefan Boltzmann
  Real 5.67e-8
End
! Ongelman ratkaisussa käytetyt kontrollitiedot.
Simulation
  Coordinate System
  String Cartesian 2D

  Simulation Type
  String Steady State

  Steady State Max Iterations
  Integer 50

  Output Intervals
  Size 1
  Integer 1

  Output File
  File "fsi.result"

  Post File
  File "fsi.ep"
End
! Virtaava aine.
Body 1
  Equation
  Integer 1

  Material
  Integer 1
End

```

```

      ! Palkki.
Body 2
  Equation
    Integer 2

  Material
    Integer 2
End
      ! Virtaavan aineen materiaaliominaisuudet.
Material 1
  Density
    Real 900

  Viscosity
    Real 0.4

  Youngs Modulus
    Variable Time
    Real Procedure "YoungYoung"

  Poisson Ratio
    Real 0.3
End
      ! Palkin materiaaliominaisuudet.
Material 2
  Density
    Real 2700

  Youngs Modulus
    Real 1.0e7

  Poisson Ratio
    Real 0.3
End
      ! Määritellään Navier-Stokes yhtälöiden ratkaisussa
      ! käytetyt menetelmät ja konvergenssitoleranssit.
Solver 1
  Equation
    String Navier-Stokes

  Stabilize
    Logical True

  Linear System Solver
    String Iterative

  Linear System Iterative Method
    String BiCGStab

```

```
Linear System Preconditioning
  String ILU1

Linear System Max Iterations
  Integer 100

Linear System Convergence Tolerance
  Real 1.0e-5

Nonlinear System Max Iterations
  Integer 10

Nonlinear System Convergence Tolerance
  Real 1.0e-5

Nonlinear System Newton After Tolerance
  Real 1.0e-5

Nonlinear System Newton After Iterations
  Integer 20

Steady State Convergence Tolerance
  Real 1.0e-4
End
  ! Elastisuusyhtälöiden ratkaisussa käytettyjen menetelmien
  ! ja konvergenssitoleranssien määrittelyt.
Solver 2
  Equation
  String Elasticity Solver

Variable
  String Displacement

Variable DOFs
  Integer 2

Procedure
  File "ElasticSolveElasticSolver"

Linear System Solver
  String Direct

Linear System Direct Method
  String Banded
```

```
Nonlinear System Newton After Tolerance
  Real 1.0e-3

Nonlinear System Newton After Iterations
  Integer 20

Nonlinear System Max Iterations
  Integer 100

Nonlinear System Convergence Tolerance
  Real 1.0e-5

Steady State Convergence Tolerance
  Real 1.0e-4
End
      ! Elementtiverkon päivityksessä käytettyjen menetelmien
      ! määrittelyt.
Solver 3
  Equation
  String Mesh Update

  Linear System Solver
  String Direct

  Linear System Direct Method
  String Banded

  Linear System Symmetric
  Logical True

  Steady State Convergence Tolerance
  Real 1.0e-4
End
      ! Kustannusfunktion laskeminen.
Solver 4
  Equation
  String CostFunction

  Procedure
  File "CostFunctionCostFunction"

  After All
  Logical True

  Steady State Convergence Tolerance
  Real 1.0
End
```

```

! Määritellään virtauskentässä ratkaistavat yhtälöt.
Equation 1
  Navier-Stokes
  Logical True

  Mesh Update
  Logical True
End
! Määritellään palkissa ratkaistavat yhtälöt.
Equation 2
  Elasticity Solver
  Logical True

  Plane Stress
  Logical True

  CostFunction
  Logical True
End
! Määritellään reunaehdot jokaisella reunalla.
Boundary Condition 1
  Target Boundaries
  Size 1
  Integer 1

  Velocity 1
  Variable Time
  Real Procedure "YoungInFlow"

  Velocity 2
  Real 0

  Mesh Update 1
  Real 0
End
Boundary Condition 2
  Target Boundaries
  Size 1
  Integer 2

  Velocity 2
  Real 0

  Mesh Update 1
  Real 0
End

```

```
Boundary Condition 3
  Target Boundaries
    Size 1
      Integer 3

  Velocity 1
    Real 0

  Velocity 2
    Real 0

  Force BC
    Logical True

  Compute Force
    Logical True

  Mesh Update 1
    Variable Displacement 1
      Real
        0 0
        1 1
      End

  Mesh Update 2
    Variable Displacement 2
      Real
        0 0
        1 1
      End
End

Boundary Condition 4
  Target Boundaries
    Size 1
      Integer 4

  Velocity 2
    Real 0

  Mesh Update 2
    Real 0
End
```


Boundary Condition 5

Target Boundaries

Size 1

Integer 5

Velocity 1

Real 0

Velocity 2

Real 0

Mesh Update 2

Real 0

End

Boundary Condition 6

Target Boundaries

Size 1

Integer 6

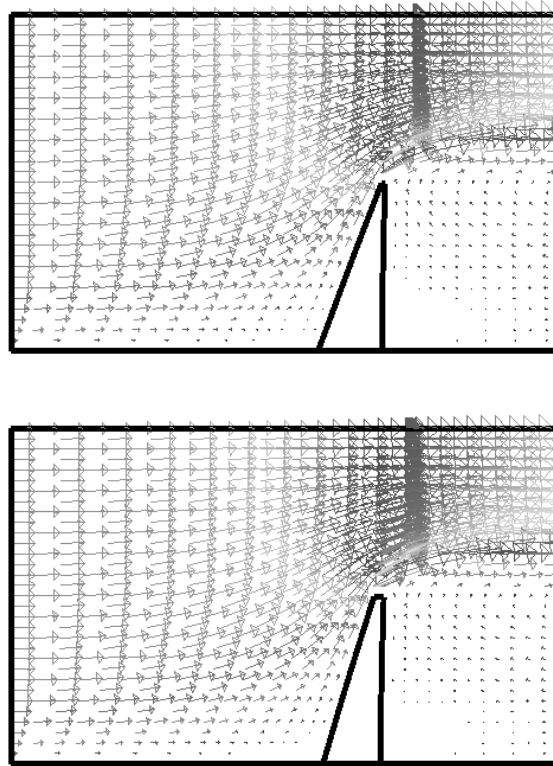
Displacement 1

Real 0

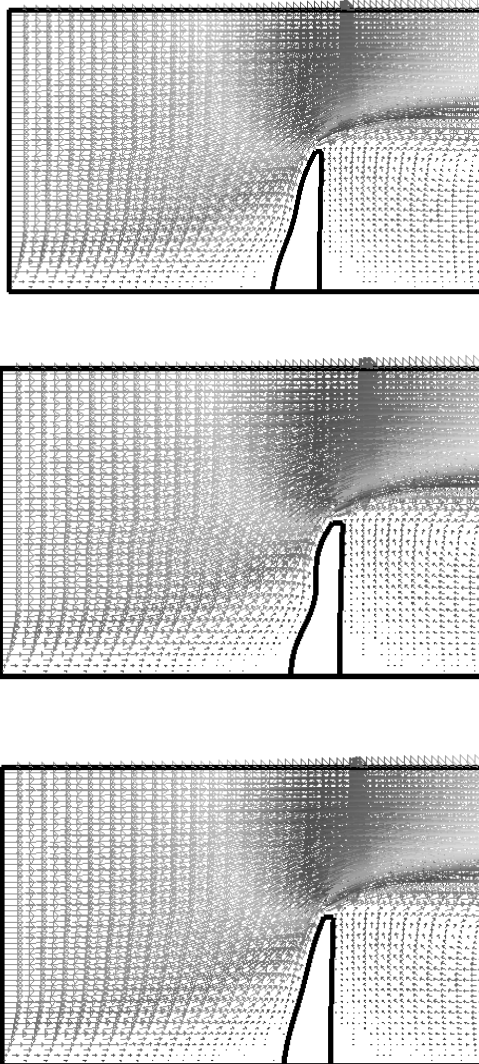
Displacement 2

Real 0

End



Kuva 15: Palkin muut potenssilaililla parametrisoidut optimimuodot. Ensimmäinen on SQP:llä ja toinen automaattisella optimoijalla saatu tulos.



Kuva 16: Palkin sinisarjan avulla löydetyt optimimuodot. Ensimmäinen simplexillä, toinen SQP:llä ja kolmas automaattisella optimoijalla saatu.