

Tampereen ammattikorkeakoulu
Rakennustekniikan koulutusohjelma
Talonrakennustekniikka
Mikko Eskola

Opinnäytetyö

Palkin laskentaohjelma

Työn ohjaaja DI Tero Markkanen
Tampere 4/2009

Mikko Eskola
Palkin laskentaohjelma
31 sivua
4/2009

Työn ohjaaja DI Tero Markkanen

Tiivistelmä

Työn lähtökohtana oli suunnitella ja toteuttaa ohjelma, jolla pystytään laskemaan erilaisista kuormituksista palkille aiheutuvia rasituksia. Erilaisia kuormitustapauksia olivat tasainen kuorma, pistekuorma, sekä pistemomentit. Lisäksi valittavana tuli olla erilaisia tuentamalleja.

Työ tehtiin rakenteiden statiikan laskennan helpottamiseksi. Statiikan laskennan tuloksia voidaan hyödyntää, kun mitoitetaan todellisia poikkileikkauksia. Ohjelmassa sovellettiin rasitusten määrittämisessä Crossin momentintasausmenetelmää.

Ohjelmasta oli tarkoitus tehdä helppokäyttöinen ja käyttäjäystävällinen. Lisäksi oli kiinnitettävä huomiota ohjelman koodin selkeyteen ja tehokkuuteen. Tarkoituksena oli, että ohjelmaa olisi myöhemmin helppo kehittää mitoittamaan eri materiaaleista koostuvia profiileita. Ohjelma toteutettiin Visual Basic 6:lla.

Avainsanat

rakenteiden statiikka, rakenteiden mekaniikka, palkki, Crossin momentintasausmenetelmä, Visual Basic, epämääräiset sauvarakenteet

Mikko Eskola

Beam calculating program

31 pages

4/2009

Thesis Supervisor

Tero Markkanen (Master of Science in Technology)

ABSTRACT

The meaning of this thesis was to design and make a program which can calculate beam stresses from different loads. Different loads were uniformly distributed load, concentrated load and concentrated moment. In addition there needed to be different bearing models available.

The work was done to ease structure calculations. These calculations can be use when the real material is designed. The Program uses Cross's moment distribution technique.

The basic idea of the program was that it should be easy to use and user friendly. In addition the programs code needed to be easy to read and efficient. The point was that the program should be easy to develop and enhance later on. The program was programmed in Visual Basic.

Keywords

construction statics, construction mechanics, beam, Cross's moment distribution, Visual Basic, bar structures

Sisällysluettelo

1	Johdanto.....	5
2	Kehitysympäristö ja työkalut	6
3	Ohjelman toiminta.....	8
3.1	Grafiikan piirto	8
3.2	Crossin momentintasausmenetelmä	10
3.2.1	Perusteet	10
3.2.2	Laskenta.....	11
4	Ohjelman testaus	18
4.1	Silmämääräisesti.....	18
4.2	Vanhojen esimerkkilaskujen avulla	19
4.3	Muiden ohjelmien avulla.....	19
5	Ohjelman käyttöliittymä.....	20
5.1	Perusikkuna	20
5.2	Tulosikkuna	21
6	Esimerkkilasku	23
7	Ohjelman jatkokehittäminen	28
7.1	Taipumanlaskenta.....	28
7.2	Mitoitus	28
7.3	Tietojen tallentaminen ja tulostaminen	28
7.4	Verkkoversio	29
8	Yhteenveto	30
	Lähteet.....	31

1 Johdanto

Tämän työn kohteena olevan palkin laskentaohjelman tarkoituksena on helpottaa ja nopeuttaa rakenteiden statiikan laskentaa. Tavoitteena on saada aikaiseksi ohjelma, jolla käyttäjä pystyy nopeasti laskemaan momentti- ja leikkausvoimakuviot antamistaan kuormituksista.

Markkinoilla on tarjolla joitakin tulevan ohjelman kaltaisia sovelluksia. Nämä ohjelmat eivät yleensä ole kovinkaan käyttäjä ystävällisiä, vaan monesti esimerkiksi yksinkertaisenkin palkin staattisten suureiden selville saamiseksi kuluu turhan paljon aikaa.

Päälähtökohtana ohjelman suunnittelussa on sovelluksen helppokäyttöisyyden takaaminen. Samalla on täytynyt pitää huolta siitä, että ohjelma on tarpeeksi monipuolinen tyydyttämään vaativimmankin käyttäjän toiveet.

Ohjelman koodista on tarkoitus tehdä dynaaminen ja selkeä kokonaisuus, ydin, johon on tulevaisuudessa helppo liittää uusia ominaisuuksia. Uusia ominaisuuksia ovat esimerkiksi eri materiaalista valmistettujen profiilien mitoitus. Aihetta on käsitelty laajemmin luvussa 7 ohjelman jatkokehittäminen.

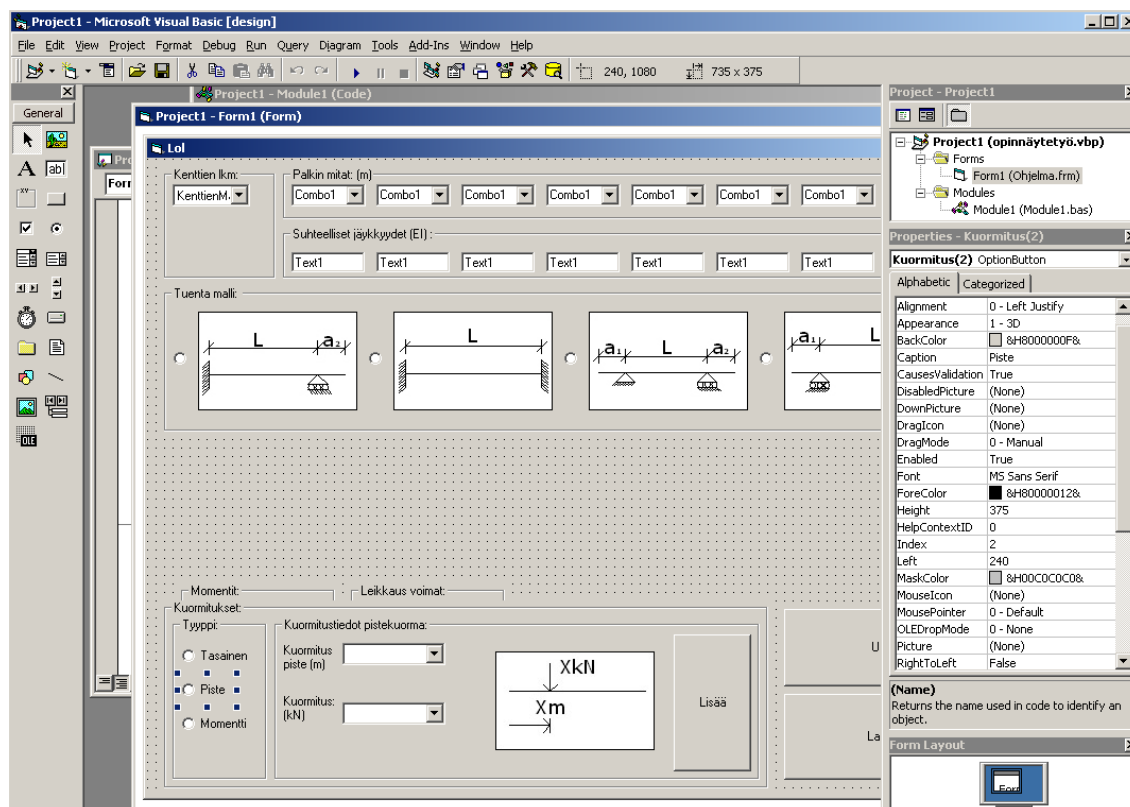
2 Kehitysympäristö ja työkalut

Ohjelmointitekniikaksi on valittavana monta eri vaihtoehtoa. Kaikista tarjolla olevista vaihtoehtoista valituksi tuli kuitenkin Microsoftin Visual Studio. Siinä on monta hyvää puolta kuten monipuolisuus, käyttäjäystävällisyys ja ennen kaikkea se on graafinen ohjelmointiympäristö.

Valituksi ohjelmointikieleksi tuli Basic, sillä raportin laatijalla oli aikaisempia kokemuksia silloisen MS-DOSin mukana tulleesta QBasic-tulkista. Vaikka QBasic eroaa Basicista monin eri tavoin, ovat ne kuitenkin toiminnaltaan ja funktioiltaan enemmän samanlaisia verrattuna vaikka oliopohjaiseen C++ kieleen.

Visual Basic 6

Visual Basic on Microsoftin kehittäämä ohjelmointikieli, josta esiteltiin ensimmäinen versio jo 1990-luvun alussa. Se on suunniteltu lähinnä ohjelmien tekemistä silmällä pitäen, mutta lisäkirjastojen ja nykyisten laajennusten avulla sillä on tehty myös grafiikaltaan näyttäviä pelejä. Visual Basic löytyy myös Microsoftin Office – tuotteiden makrokielenä. Kuviossa 1 on näkymä Visual Studio 6 työtilasta.



Kuvio 1: Näkymä Visual Studion työtilasta

Huonona puolena Visual Basicissa on se, että sillä tehdyt ohjelmat toimivat ainoastaan Windows-käyttöjärjestelmissä. Kuitenkin esimerkiksi Linuxille on kehitetty monenlaisia Windows-emulaattoreita, joiden avulla ohjelmia pystytään ajamaan myös Linux-käyttöjärjestelmissä.

3 Ohjelman toiminta

3.1 Grafiikan piirto

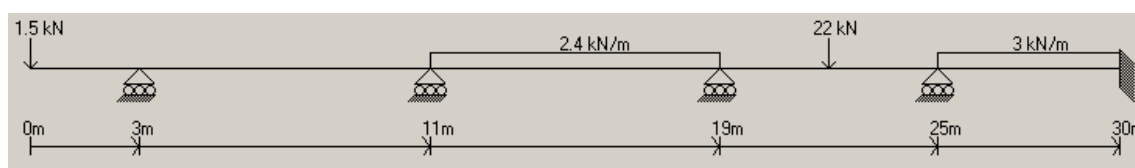
Ohjelman muuttuvan grafiikan piirroksessa käytetään Visual Basicin omia grafiikkakomentoja kuten: *Line* ja *Circle*. Tekstien paikkojen määrittely tapahtuu puolestaan komennoilla *CurrentY* ja *CurrentX*. Valmiit kuvat ovat upotettu ohjelmaan ohjelman *Picture*-ominaisuudella.

Muuntokerroin

Koska palkin pituus ruudulla on aina sama, 700 pikseliä, joudutaan tukien, kuormien ja mittaviivojen piirtämistä varten laskemaan muuntokerroin. Se lasketaan jakamalla ruudulla oleva palkinpituus todellisella palkin pituudella. Samoin menetellään myös momentti- ja leikkausvoimakuvioita piirrettäessä: itseisarvoiltaan isoimpien rasitusten perusteella lasketaan muuntokertoimet, joiden avulla kuvaajat saadaan sovitettua niille varattuun tilaan.

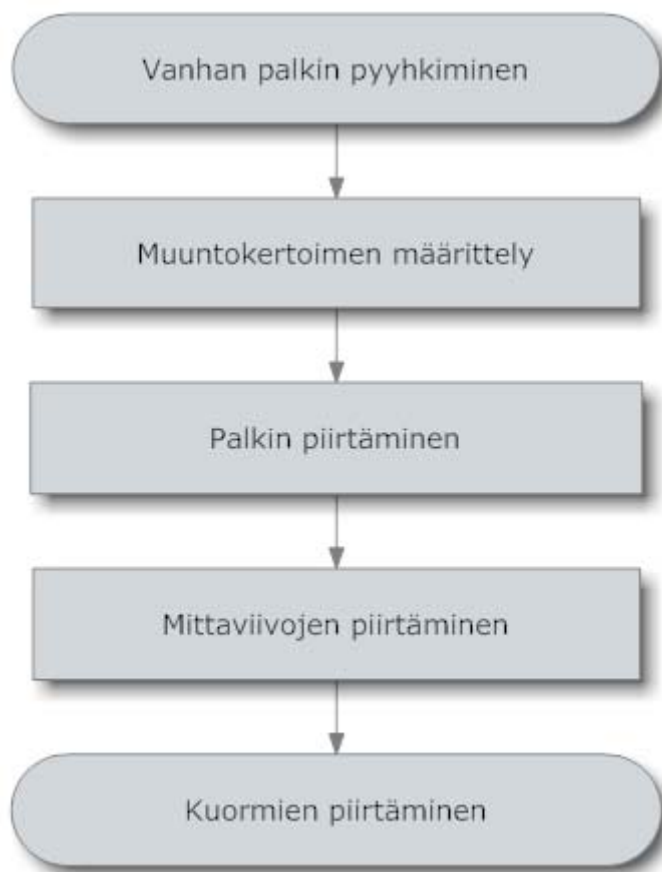
Pääsivu

Ohjelman pääsivulla oleva palkki (kuvio 2) piirretään aina, kun käyttäjä antaa uusia tai vaihtaa vanhoja arvoja.



Kuvio 2: Ohjelman pääsivulle piirretty palkki

Arvojen vaihtuessa vanha palkki kuormituksineen pyyhitään pois ruudulta *Cls*-komennolla. Tämän jälkeen ohjelma suorittaa kaikki tarpeelliset aliohjelmat, joilla uusi palkki sekä kuormitukset saadaan piirrettyä. Piirtäminen etenee kuvion 3 mukaisesti.



Kuvio 3: Grafiikan piirtämisen eteneminen

Tulossivu

Momentti- ja leikkausvoimakuviot piirretään kukin samalla logiikalla. Sen mukaan Crossin momentintasausmenetelmää hyödyntämällä ohjelma on laskenut ja taulukoinut momentin ja leikkausvoiman palkin jokaiselle millimetrimille. Kuvaajat piirretään käymälä taulukko läpi *For*-silmukan avulla kuvion 4 esittämällä tavalla.

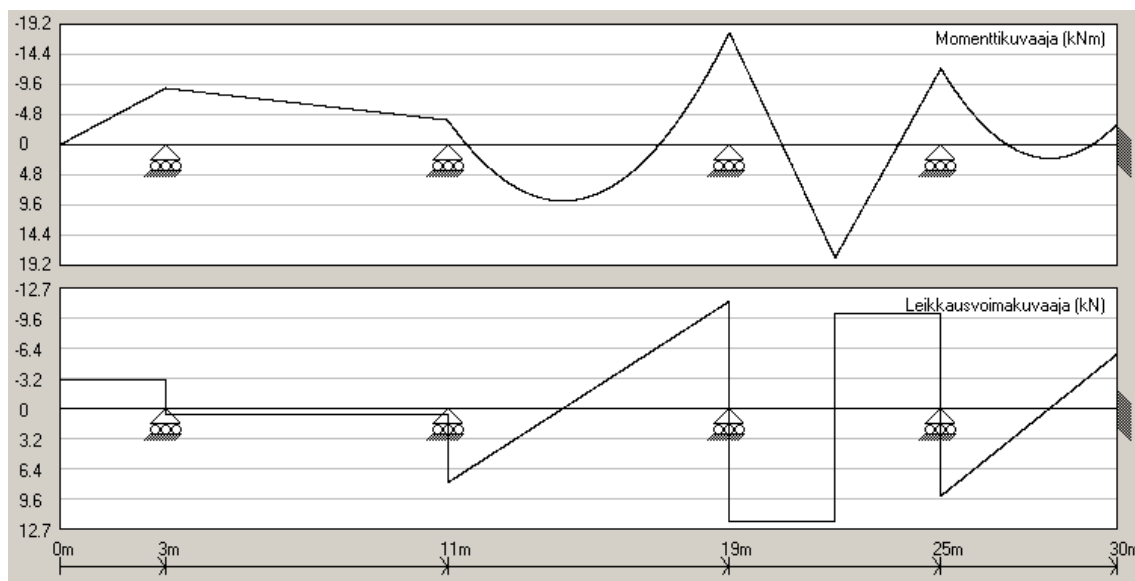
```

For i = 0 To PalkinPituusMm - 5 Step 5
    Me.Line (i * MuuntoKerroin + ViivanX1, MomenttiTaul(i) * MuuntoKerroinY + 100) -
    ((i + 5) * MuuntoKerroin + ViivanX1, MomenttiTaul(i + 5) * MuuntoKerroinY + 100)
Next

```

Kuvio 4: Momenttikuvaajan piirto koodi

ViivanX1 lisätään X-koordinaatin arvoon joka kerta, sillä kuvaaja ei lähde ruudun reunaan. Samoin perustein lisätään luku 100 joka kerta Y-koordinaatin arvoon. Kuvion 2 tapauksesta piirretyt momentti- ja leikkausvoimakuvaajat näkyvät kuviossa 5.



Kuvio 5: Momentti- ja leikkausvoimakuvajat

3.2 Crossin momentintasausmenetelmä

3.2.1 Perusteet

Momentintasausmenetelmän on esittänyt vuonna 1932 Hardy Cross, jonka mukaan se on saanut nimensä.

”Crossin menetelmä perustuu siirtymämenetelmän periaatteisiin. Se on likiarvomenetelmä, jossa peräkkäisillä laskutoimituksilla (momentintasauksilla) lähestytään tarkkaa ratkaisua ja laskenta voidaan lopettaa kun on saavutettu haluttu tarkkuus.” (Iivonen 2004, 48)

Käsin laskettuna Crossin momentintasausmenetelmä on pitkä ja monivaiheinen urakka, varsinkin jos kenttien lukumäärä on suurempi kuin kolme. Näin ollen myös virheiden mahdollisuus kasvaa suureksi ja virhe alussa tulee pilaamaan pitkän työrupeaman.

Vaikka Crossin momentintasausmenetelmä on niin sanottu likiarvomenetelmä, saadaan tietokoneella laskettua tulokset sekunnin murto-osassa, vaikka kymmenen merkitsevän numeron tarkkuudella. Useimmiten kuitenkin jo parin desimaalin tarkkuus on aivan riittävä käytännön laskennan kannalta.

3.2.2 Laskenta

Crossin momentintasausmenetelmään liittyy tiettyjä muuttujia, jotka pitää laskea ennen varsinaista momenttientasausta. Lisäksi tarvitaan kiinnitysmomenttien laskentaan eri kuormitustapauksille erinäisiä kaavoja, joita löytyy mm. *Loikkasen (1975, 354)* kirjasta.

Tässä työssä ei valoteta menetelmän perusteena olevaa teoriaa. Eikä myöskään sitä kuinka Crossin momentintasausmenetelmää käytetään, vaan selostetaan lähinnä miten se on ohjelmoitu koneen laskettavaksi. Kyseisen menetelmän teorian opettelemiseen tarkoitettua hyvää materiaalia löytyy esimerkiksi aikaisemmin mainituista *Iivosen* ja *Loikkasen* teoksissa.

Jäykkyysluvut

Ohjelma laskee jäykkyysluvut käyttäjän antamien suhteellisten jäykkyyksien ja kenttämittojen jakona. Kuvion 2 jäykkyysluvut lasketaan kuvion 6 koodilla.

```
If Val(UlokeMitta(1)) <= 0 Then
    JaykkyysLuvut(1) = 0.75 * KenttaJaykkyys(1) / KenttaMitta(1)
Else
    JaykkyysLuvut(1) = KenttaJaykkyys(1) / KenttaMitta(1)
End If

If KenttienMaara > 1 Then
    For i = 2 To KenttienMaara
        JaykkyysLuvut(i) = KenttaJaykkyys(i) / KenttaMitta(i)
    Next
End If
```

Kuvio 6: Jäykkyyslukujen laskenta kuvion 2 tapaukselle

Jakoluvut

Jakoluvut lasketaan tuen molemmille puolille jakamalla kentän jäykkyysluku kentän jäykkyysluvun ja viereisen kentän jäykkyysluvun summalla. Kuvion 7 koodi laskee jakoluvut jokaiselle tuentamallille.

```
For i = 1 To (KenttienMaara - 1)
    JakoLuvut(i, 2) = -JaykkyysLuvut(i) / (JaykkyysLuvut(i) + JaykkyysLuvut(i + 1))
    JakoLuvut(i + 1, 1) = -JaykkyysLuvut(i + 1) / (JaykkyysLuvut(i) + JaykkyysLuvut(i + 1))
Next
```

Kuvio 7: Jakolukujen laskenta

Jakolukujen taulukoinnissa ensimmäiseen arvoon syötetään tuen numero ja toiseen arvoon se kuuluuko luku tuen oikealle vai vasemmalle puolelle. Numerointi kulkee vasemmalta oikealle.

Siirtoluvut

Siirtoluvuiksi tulee kaikille tuille paitsi reunanivelille arvoksi 0,5. Siirtoluvulla siirtomomentti siirretään viereiselle tuelle. Reunanivelille siirtomomentti on nolla, sillä reunanivel ei vastaanota momenttia.

Lähtömomentit

Lähtömomentit lasketaan erikseen jokaiselle sauvalle ajatellen sauvanpäihin jäykät tuet, poikkeuksena reunoilla olevat rullatuet, jotka eivät ota momenttia. Esimerkiksi kuvion 2 välin 11-19m sauvalle lähtömomentit lasketaan kaavoilla 1 ja 2.

Tasaisen kuorman aiheuttamat lähtömomentit lasketaan seuraavilla kaavoilla.

$$MK_1 = -\frac{qs}{12L^2} [12ab^2 + s^2(L - 3b)] \quad (1)$$

$$MK_2 = \frac{qs}{12L^2} [12a^2b + s^2(L - 3a)] \quad (2)$$

jossa

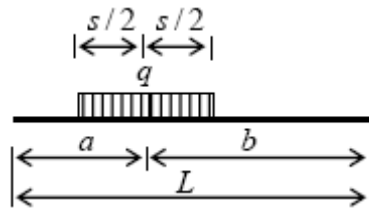
q on tasainen kuorma

s on kuormanpituus

a on etäisyys vasemmalta tuelta kuorman keskipisteeseen

b on etäisyys oikealta tuelta kuorman keskipisteeseen ja

L on sauvan pituus, kuvion 8 mukaan.



Kuvio 8: Tasaisen kuorman aiheuttamien lähtömomenttien laskemiseen käytettävät suureet

Kuviossa 9 on esillä pala koodia, jolla lasketaan kuvion 2 välin 11-19m tasaisen kuorman aiheuttamat lähtömomentit. Pisteiden paikalla on itse ohjelmassa monta riviä koodia. Tämä johtuu siitä, että jos kuormitus jatkuisi jommankumman tuen ylitse, arvot s , a ja b tulisi laskea eri tavalla.

```

For i = 2 To (Val(KenttienMaara)) 'kenttien läpikäynti
  For ii = 1 To TsKuormaLaskuri 'kuormienläpikäynti
    'jos kuorma on kentän sisässä
    If TsKuormaX1(ii) < JuoksevaKenttaMitta(i) And TsKuormaX2(ii) < JuoksevaKenttaMitta(i) _
      And TsKuormaX1(ii) > (JuoksevaKenttaMitta(i - 1)) Then
      s = TsKuormaX2(ii) - TsKuormaX1(ii)
      a = TsKuormaX1(ii) - (JuoksevaKenttaMitta(i - 1)) + s / 2
      b = CSng(KenttaMitta(i)) - a
      l = CSng(KenttaMitta(i))
      TsKuormaMom(i, 1) = -(((TsKuorma(ii) * s) / (12 * l ^ 2)) * (12 * a * b ^ 2 + s ^ 2 * (1 - 3 * b)))
      TsKuormaMom(i, 2) = (((TsKuorma(ii) * s) / (12 * l ^ 2)) * (12 * a ^ 2 * b + s ^ 2 * (1 - 3 * a)))
      .
      .
      .
      TsKuormaMomSumma(i, 1) = TsKuormaMomSumma(i, 1) + TsKuormaMom(i, 1)
      TsKuormaMomSumma(i, 2) = TsKuormaMomSumma(i, 2) + TsKuormaMom(i, 2)
      TsKuormaMom(i, 1) = 0 'nollataan muuttujat
      TsKuormaMom(i, 2) = 0
    Next
  Next
Next

```

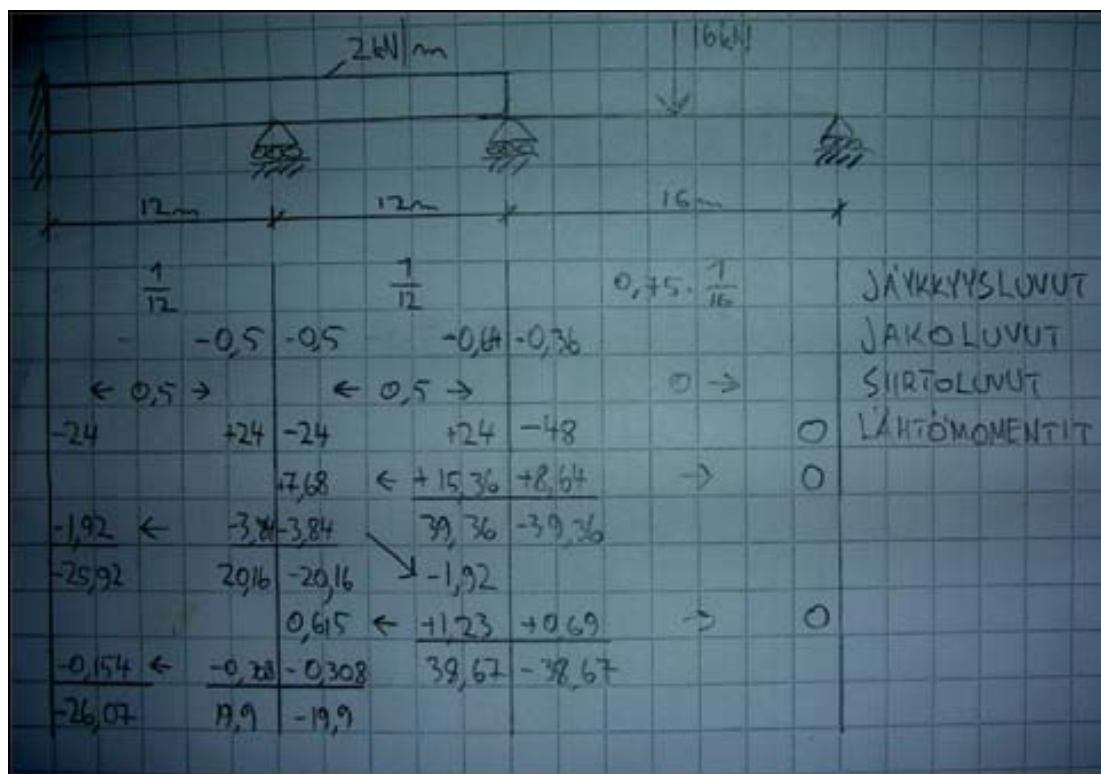
Kuvio 9: Osa lähtömomenttien laskentakoodista

Kuvion 9 koodi käy lävitse jokaisen palkin kentän, lukuun ottamatta mahdollista uloketta ja ensimmäistä kenttää, sekä jokaisen käyttäjän syöttämän tasaisen kuorman. Ulokkeen ja ensimmäisen kentän lähtömomenttien määrittämisessä käytetään eri koodia johtuen niiden riippuvuudesta toisiinsa. Kuormansijaintivaihtoehdot läpikäytyään lähtömomentit summataan TsKuormaMomSumma-aulukkkoon. Taulukoinnissa ensimmäinen numero vastaa tuen numeroa ja toinen sitä, onko kyseessä vasen vai oikea puoli tuesta?

Crossaus

Kun lähtömomentit on saatu laskettua, voidaan aloittaa itse crossaus. Se alkaa tuelta, jolla on itseisarvoltaan isoin lähtömomenttien summa. Saatu arvo kerrotaan kummankin puolen jakoluvulla. Tämän jälkeen saadut tulokset kerrotaan siirtoluvuilla ja summataan

viereisen tuen lähtömomenttiin. Jakoluvulla kerrottu lähtömomentti summataan myös alkuperäiseen lähtömomenttiin. Tämän jälkeen haetaan jälleen itseisarvoltaan isoin lähtömomenttien summa. Operaatiota toistetaan niin kauan, kunnes päästään haluttuun tarkkuuteen. Toisin sanoen sitä jatketaan niin kauan, kunnes lähtömomenttien summa saavuttaa halutun raja-arvon. Kuviossa 10 on esimerkki crossauksesta käsilaskentana.



Kuvio 10: Crossausta käsinlaskentana

Ohjelma suorittaa crossauksen kuvion 11 koodin mukaan.

```

Do While Abs(Isoin) > 0.001
    Isoin = 0
    For i = 2 To (Val(KenttienMaara)) 'momenttien yhdistäminen
        YhdMom(i) = Momentit(i - 1, 2) + Momentit(i, 1)
    Next
    For i = 2 To (Val(KenttienMaara))
        If Abs(Isoin) < Abs(YhdMom(i)) Then
            Isoin = YhdMom(i)
            Paikka = i
        End If
    Next
    If Isoin = 0 Then
        Exit Do
    End If
    'vasemmalle siirtyvä momentti
    SiirMom(1, 2) = Isoin * JakoLuvut(Paikka - 1, 2)
    'oikealle siirtyvä momentti
    SiirMom(1, 1) = Isoin * JakoLuvut(Paikka, 1)
    'siirretään momentti vasemmalle
    Momentit(Paikka - 1, 1) = Momentit(Paikka - 1, 1) + SiirtoLuvut(Paikka - 1, 2) * SiirMom(1, 2)
    'siirretään momentti oikealle
    Momentit(Paikka, 2) = Momentit(Paikka, 2) + SiirtoLuvut(Paikka, 1) * SiirMom(1, 1)
    'summataan siirto momentti alkuperäiselle tuelle
    Momentit(Paikka - 1, 2) = Momentit(Paikka - 1, 2) + SiirMom(1, 2)
    Momentit(Paikka, 1) = Momentit(Paikka, 1) + SiirMom(1, 1)
    DoEvents
Loop

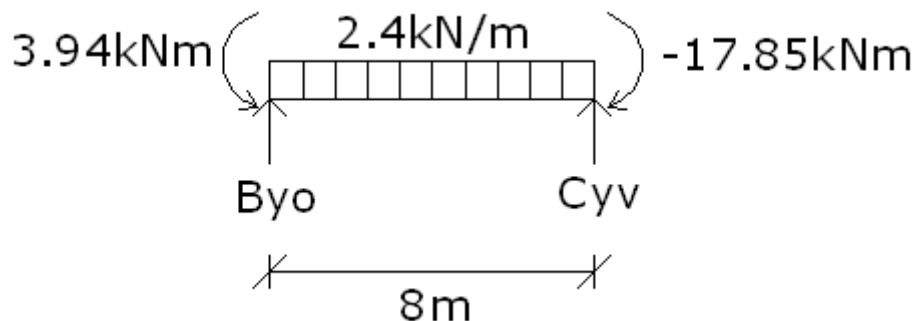
```

Kuvio 11: Crossauksen koodi

Ohjelma toistaa crossaus-silmukkaa niin kauan, kunnes isoin tasattava momentti on pienempi kuin 0,001. Arvo 0,001 vastaa todellisuudessa 0,001kNm:ä.

Tukireaktioiden laskeminen

Kun on crossattu, jaetaan palkki osapalkkeihin, joka tapahtuu tukien kohdalta. Kuviossa 12 on piirretty kuvion 2 välin 11-19m osapalkin vapaakappalekuva.



Kuvio 12: Vapaakappalekuva, jossa on ratkaistuna lopulliset sauvanpäämomentit.

Tukivoimat Byo ja Cyv ratkaistaan kuviossa 12 normaalisti statiikan tasapainoehtojen mukaan. Kuviossa 13 on esimerkkinä pala koodista, joka hakee osapalkissa vaikuttavat

pistekuormat ja laskee niistä aiheutuvat momentit sauvan päihin. Samoin toimitaan myös muiden kuormitustapausten kanssa. Lopuksi saadut momentit summataan ja jaetaan osapalkin pituudella.

```

.
.
.
'kentässä vaikuttavat pistekuormat
  For ii = 1 To PsKuormaLaskuri 'kuormienläpikäynti
    'jos pistekuorma on kentän sisässä
    If PsKuormaX(ii) > JuoksevaKenttaMitta(i - 1) And _
      PsKuormaX(ii) < JuoksevaKenttaMitta(i) Then
      a = PsKuormaX(ii) - (JuoksevaKenttaMitta(i - 1))
      b = Val(KenttaMitta(i)) - a
      PsKuormaMomOik = PsKuormaMomOik + PsKuorma(ii) * b
      PsKuormaMomVas = PsKuormaMomVas - PsKuorma(ii) * a
    End If
  Next
'mom oik puol tuen ymp ratk Ay
TukiVoimat(i, 1) = (-Momentit(i, 1) + TsKuormaMomOik + MomKuormaMomOik + _
PsKuormaMomOik - Momentit(i, 2)) / Val(KenttaMitta(i))
'mom vas puol tuen ymp ratk By
TukiVoimat(i, 2) = -(-Momentit(i, 1) + TsKuormaMomVas + MomKuormaMomVas + _
PsKuormaMomVas - Momentit(i, 2)) / Val(KenttaMitta(i))
.
.
.

```

Kuvio 13: Koodia tukivoimien laskennasta

Momentti- ja leikkausvoimakuvioiden piirto

Momentti- ja leikkausvoimakuvioiden piirron perustana olevat logiikat ovat hyvin samanlaiset, joten idean selvittämisessä käytetään pelkästään momenttikuvaajan piirrosta tarvittavia asioita.

Ennen piirtämisen aloittamista ohjelma laskee ja taulukoi momentin palkin jokaista millimetriä kohden. Laskeminen tehdään erikseen jokaiselle kuormitustapaukselle. Kuviossa 14 on koodi tukivoimista aiheutuvien momenttien taulukointiin. Kuvion 15 koodi puolestaan yhdistää eri kuormitustapauksista aiheutuvat momentit toisiinsa.

```

For i = 1 To PalkinPituusMm
    .
    .
    .
    'Tukivoimista aiheutuva momentti pisteessä i
    For ii = 0 To (Val(KenttienMaara) + 1)
        If TvMomenttiX(ii) < i Then
            TvMomenttiTaul(i) = TvMomenttiTaul(i) + (YhdTukiVoimat(ii + 1) _
                * (i - TvMomenttiX(ii))) * 10 ^ -6
        End If
    Next
Next

```

Kuvio 14: Tukivoimista aiheutuvien momenttien taulukointia

```

For i = 1 To PalkinPituusMm
    MomenttiTaul(i) = TsMomenttiTaul(i) + MomMomenttiTaul(i) _
        + PsMomenttiTaul(i) + TvMomenttiTaul(i) + Momentit(1, 1)
Next

```

Kuvio 15: Eri kuormitusten aiheuttamien momenttien summaus

Kun momentit on saatu laskettua ja taulukoitua koko palkin pituudelta, voidaan piirtää itse kuvaaja. Piirtämisestä on kerrottu osiossa 3.1 Grafiikan piirto.

4 Ohjelman testaus

Tärkeänä osana jokaisen ohjelman kehitystä on sen testaus, varsinkin tällaisessa tapauksessa, kun ohjelmaa tullaan käyttämään apuna rakenteiden mitoituksessa. Pahimmillaan väärän tuloksen perusteella mitoitettu rakenne voi sortua ja aiheuttaa suurta vaaraa.

Testaustulosten luotettavuuden vuoksi on valittu kolme eri menetelmää, joilla ohjelman laskentaa on testattu. Käytännössä testaus on suoritettu laskemalla ohjelmalla useita erilaisia ja eri lailla kuormitettuja palkkeja. Tämän jälkeen tuloksia on analysoitu seuraavaksi esitetyillä menetelmillä.

4.1 Silmämääräisesti

Silmämääräinen testaus on nopea keino saada tietoa, laskeeko ohjelma oikein vai ei. Seuraavissa luetelmissa on esitetty silmämääräisen testauksen perusperiaatteet.

Silmämääräinen testaus perustuu

- kokemukseen palkkien käyttäytymisestä
- erilaisten kuormien ja rasiutusten aiheuttamiin vaikutuksiin
- momentti- ja leikkausvoimakuvioiden yhteneväisyyksiin.

Silmämääräisessä tarkastelussa kiinnitetään huomiota seuraaviin asioihin:

- Momentin ääriarvot ovat usein leikkausvoiman nollakohdassa.
- Momenttikuvaaja on paraabelin muotoinen tasaisten kuormien vaikutusalueella.
- Leikkausvoima kuvaaja on laskeva tai nouseva suora tasaisten kuormien vaikutusalueella.
- Pistekuorma aiheuttaa hyppäyksen leikkausvoimakuvaajaan.
- Pistementti aiheuttaa hyppäyksen momenttikuvaajaan.
- Tukemattoman palkinpään momentin tulee olla nolla.
- Rullatuellisen palkinpään momentin tulee olla nolla.
- Vastaavatko rasiutukset suuruudeltaan syötettyjä kuormia?

4.2 Vanhojen esimerkkilaskujen avulla

Tampereen ammattikorkeakoulun rakenteiden mekaniikan tunneilla on laskettu useita erilaisia palkkeja erilaisilla kuormitusyhdistelmillä. Laskemalla ohjelmalla näitä tehtäviä uudestaan ja vertaamalla saatuja tuloksia on todettu ohjelman olevan näiltä osin kunnossa. Lisäksi ohjelmaa on testattu mm. *Loikkasen* kirjan laskuesimerkkien avulla.

4.3 Muiden ohjelmien avulla

Markkinoilla on myös joitakin muita statiikanlaskennassa ja mitoituksessa käytettäviä ohjelmia. Tampereen ammattikorkeakoululla on lisenssit mm. DOF-Puuhun ja PUPA-Xiin, joiden avulla ohjelmaa on testattu.

Vertailua varten laskettiin useita erilaisia esimerkkitapauksia, kaikilla kolmella ohjelmalla. Vertailussa saadut tulokset vastasivat käytännössä toisiaan. DOF-Puuhun verrattuna ohjelma antoi sen kanssa lähes identtisiä tuloksia. Verrattaessa tuloksia PUPAXilla saatuihin ne erosivat hieman toisistaan. Käytännön mitoituksen kannalta tulosten eroavaisuudella ei olisi ollut juurikaan merkitystä.

Vaikka otanta ei ollut kovinkaan suuri, antaa se kuitenkin hyvän kuvan siitä, että ohjelma laskee oikein. Pistemomenttien laskennan testauksessa jouduttiin turvautumaan pelkkään DOF-Puuhun sillä PUPAX ei tue pistemomentteja.

5 Ohjelman käyttöliittymä

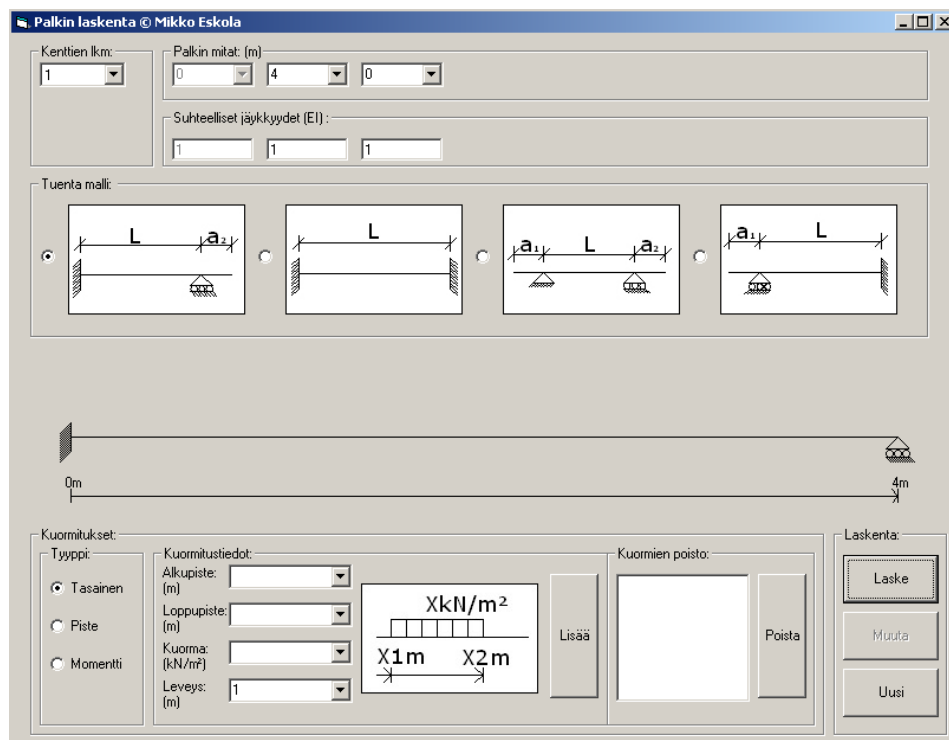
Tutkimalla ja käyttämällä muita mitoitusohjelmia huomaa niiden suurimmaksi ongelmaksi käytettävyyden. Suurin osa ongelmista liittyi liian monitasoiseen käyttöliittymään. Ohjelmat upottivat turhan usein käyttäjänsä ikkunaviidakkoon, jonne kokeneempikin koneenkäyttäjä helposti hukkui.

Toinen huomattava hämmennyksen aihe oli eri yksiköiden sekoittaminen. Ensimmäisessä ikkunassa pituuden yksikkönä saattoi olla metri, toisessa taas millimetri. Huolimattomasti syötettyihin arvoihin saattoi siis tulla tuhatkertainen virhe.

Ohjelman käyttöliittymää tehdessä ja suunnitellessa pyrittiin pitämään helppokäyttöisyys tärkeimpänä ohjenuorana. Ideat silmille pomppivista kysymysruuduista tai monijakoisista ja mutkikkaista valintaikkunoista unohdettiin välittömästi. Tavoitteena oli saada pääsivulle kaikki syöttöruudut ja napit mitä ohjelma tulee tarvitsemaan.

5.1 Perusikkuna

Ohjelman käynnistyksen jälkeen avautuu kuvion 16 mukainen perusikkuna.



Kuvio 16: Ohjelman perusikkuna

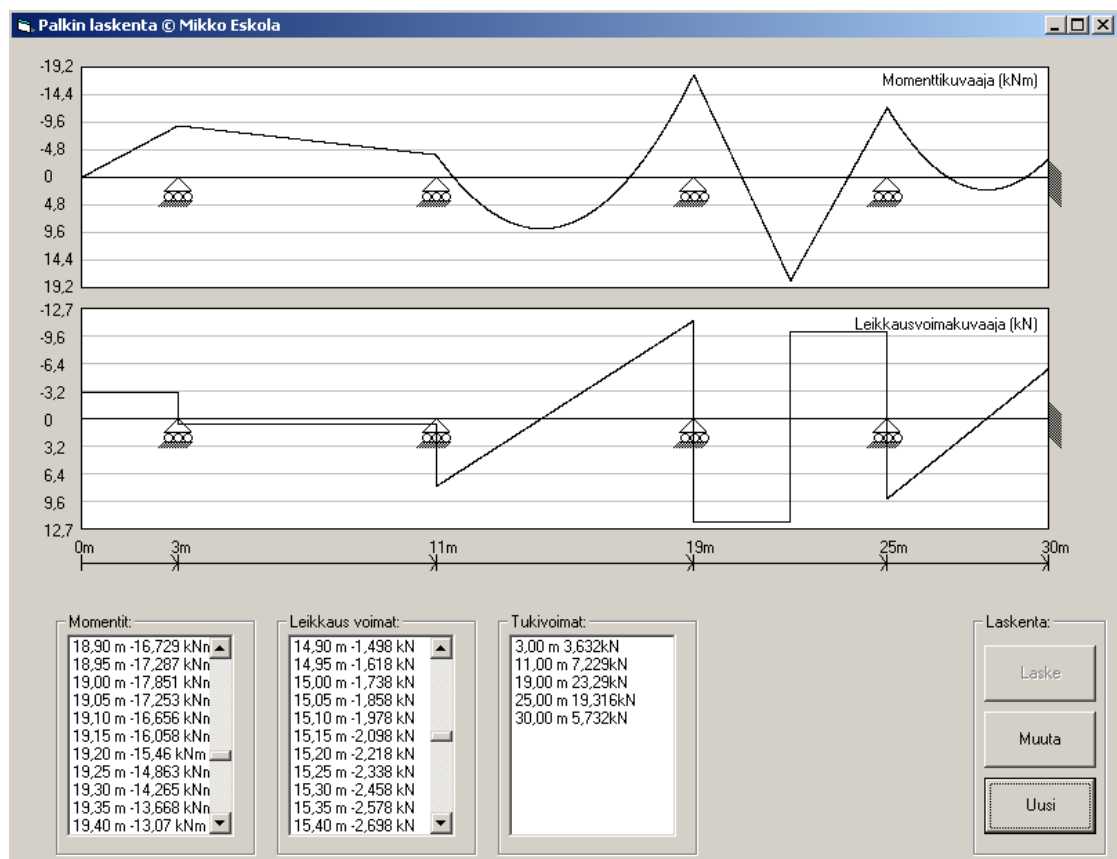
Perusikkunan kautta käyttäjä hallinnoi

- kenttien lukumäärää
- palkin mittoja
- suhteellisia jäykkyyksiä
- tuentamalleja
- kuormituksia.

Lisäksi on myös mahdollista poistaa jo syötettyjä kuormia, *kuormien poisto* -kohdan avulla. Painamalla nappulaa *Uusi* ohjelma nollaa tehdyt muutokset ja palkin suunnittelun voi aloittaa alusta.

5.2 Tulosikkuna

Tulosikkuna tuo ruudulle kuvion 17 mukaiset kuvaajat ja tulokset.



Kuvio 17: Ohjelman tulosikkuna

Tulosikkunan kautta käyttäjä näkee

- palkin geometrian
- momenttikuvaajan
- leikkausvoimakuvaajan
- momentit 5 cm:n välein
- leikkausvoimat 5 cm:n välein
- tukivoimat.

Tulosikkuna koostaa ruudulle lasketut rasitukset käyttäjän tarkasteltavaksi. Tyhjään tilaan on suunnitteilla mitoitusvalikko. Sen avulla käyttäjä pystyy valitsemaan ja mitoittamaan lasketun palkin puu-, teräs- tai teräsbetonipoikkileikkauksena. Asiasta on kerrottu lisää luvussa 7 ohjelman jatkokehittäminen.

6 Esimerkkilasku

Tässä ohjelmaan tutustuttava esimerkkilasku on suunniteltu siten, että sen avulla käyttäjä omaksuu ohjelman kaikki ominaisuudet. Näytöllä oleviin teksteihin tai syöttökenttiin viitattaessa on käytetty *kursivoitua* tyyliä.

Kenttien lukumäärä

Käyttäjä valitsee haluamansa kenttien lukumäärän *Kenttien lkm:* -valikosta kuvion 18 mukaan.

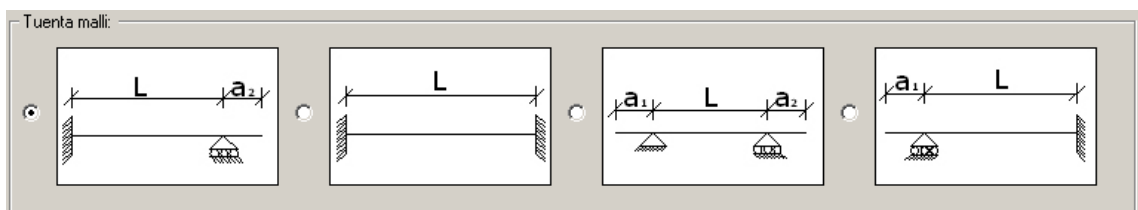


Kuvio 18: Kenttien lukumäärän valitseminen

Esimerkissä valitaan kenttien lukumääräksi neljä.

Tuentamalli

Käyttäjällä on mahdollisuus valita *Tuentamalli:* neljästä eri vaihtoehdosta kuvion 19 mukaan.

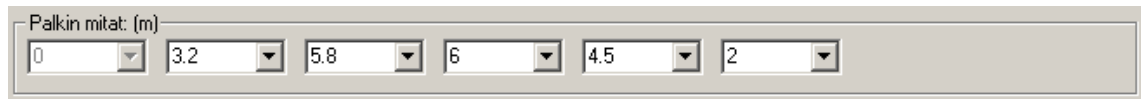


Kuvio 19: Tuentamallin valitseminen

Esimerkissä valitaan tuentamalliksi kuvion 19 osoittama tapaus, joka on kuviossa vasemmanpuoleisin.

Palkin mitat

Käyttäjä asettaa palkille haluamansa pituuden *Palkin mitat: (m)* -kehyksestä. Mitat annetaan metreinä. Desimaalierottimena voidaan käyttää pilkkua ja pistettä. Kuviossa 20 näkyvät mitat, jotka syötetään esimerkki laskuun.



Palkin mitat: (m)

0	3.2	5.8	6	4.5	2
---	-----	-----	---	-----	---

Kuvio 20: Palkin pituuden syöttö

Suhteelliset jäykkyydet

Jos mitoitettavan palkin jäykkyys vaihtelee, voidaan kuvion 21 mukaisiin ruutuihin antaa kyseisen sauvan suhteellinen jäykkyys. Esimerkissä ratkaistaan tasajäykkä palkki, joten jokaiseen ruutuun jätetään ykkönen.



Suhteelliset jäykkyydet (EI):

1	1	1	1	1	1
---	---	---	---	---	---

Kuvio 21: Suhteellisten jäykkyyksien syöttö

Rakennemalli

Ruudun keskiosaan automaattisesti päivittyvä rakennemalli näyttää käyttäjälle palkin mitat, tuet sekä syötetyt kuormitukset. Esimerkkiä seuraamalla palkin pitäisi olla kuvion 22 mukainen.



Kuvio 22: Rakennemalli

Kuormitukset

Kuormitukset: -kehyksestä syötetään palkille kuormia, joita ovat: *Tasainen*, *Piste* ja *Momentti*. Haluttu kuormitus valitaan *Tyyppi:* -kehyksestä, minkä jälkeen *Kuormitustiedot:* -kehyksestä päästään antamaan kuormitustietoja. Valitaan ensimmäisenä *Tasainen* kuvion 23 mukaan.

Tyyppi:

☒ Tasainen

☐ Piste

☐ Momentti

Kuvio 23: Kuormitus tyyppin valinta

Kuormitustiedot: -kehykseen annetaan kuvion 24 mukaiset arvot ja painetaan *Lisää* -nappulaa.

Kuormitustiedot:			Kuormien poisto:	
Alkupiste: (m)	0		1: 0-17,5m 4,2kN/m Poista	
Loppupiste: (m)	17,5			
Kuorma: (kN/m²)	4,2			
Leveys: (m)	1			

Kuvio 24: Tasaisen kuorman syöttö

Kuorma ilmestyy rakennemallin lisäksi myös *Kuormien poisto:* -ikkunaan. Kuorma poistetaan valitsemalla haluttu kuorma *Kuormien poisto:* -ikkunasta ja painamalla *Poista* -nappulaa.

Lisää -nappulan vasemmalla puolella on kuva, joka selkeyttää kuormitustietojen antamista. Antamalla miinusmerkkisen arvon kohtaan *Kuorma: (kN/m²)*, kuorma suuntautuu alhaalta ylöspäin. Miinusmerkkisen arvon antamisella voidaan myös vaihtaa pistekuorman ja pistemomentin suuntaa.

Pistekuormien syöttöön pääsee klikkaamalla kuvion 23 esittämästä valinnasta kohtaa *Piste*. Esimerkissä annetaan *Kuormitustiedot:* -kehykseen pistekuormille kuvion 25 esittämät arvot.

Kuormitustiedot:

Kuormitus piste (m)

Kuormitus: (kN)

XkN

Xm

Lisää

Kuormien poisto:

1: 5,8m 12kN
2: 21,5m 4kN

Poista

Kuvio 25: Pistekuorman syöttö

Lisätään palkille vielä pistemomentti kuvion 26 esittämällä arvoilla.

Kuormitustiedot:

Sijainti: (m)

Momentti: (kNm)

$XkNm$

Xm

Lisää

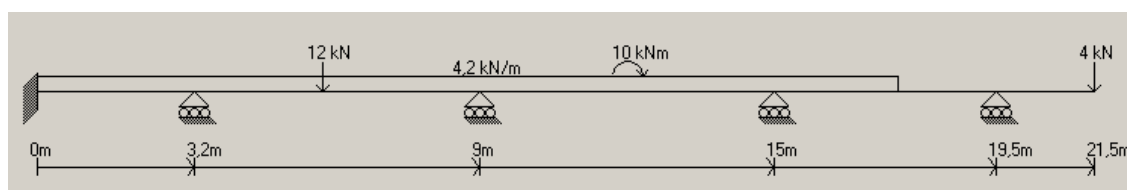
Kuormien poisto:

1: 12m 10kNm

Poista

Kuvio 26: Pistemomentin syöttö

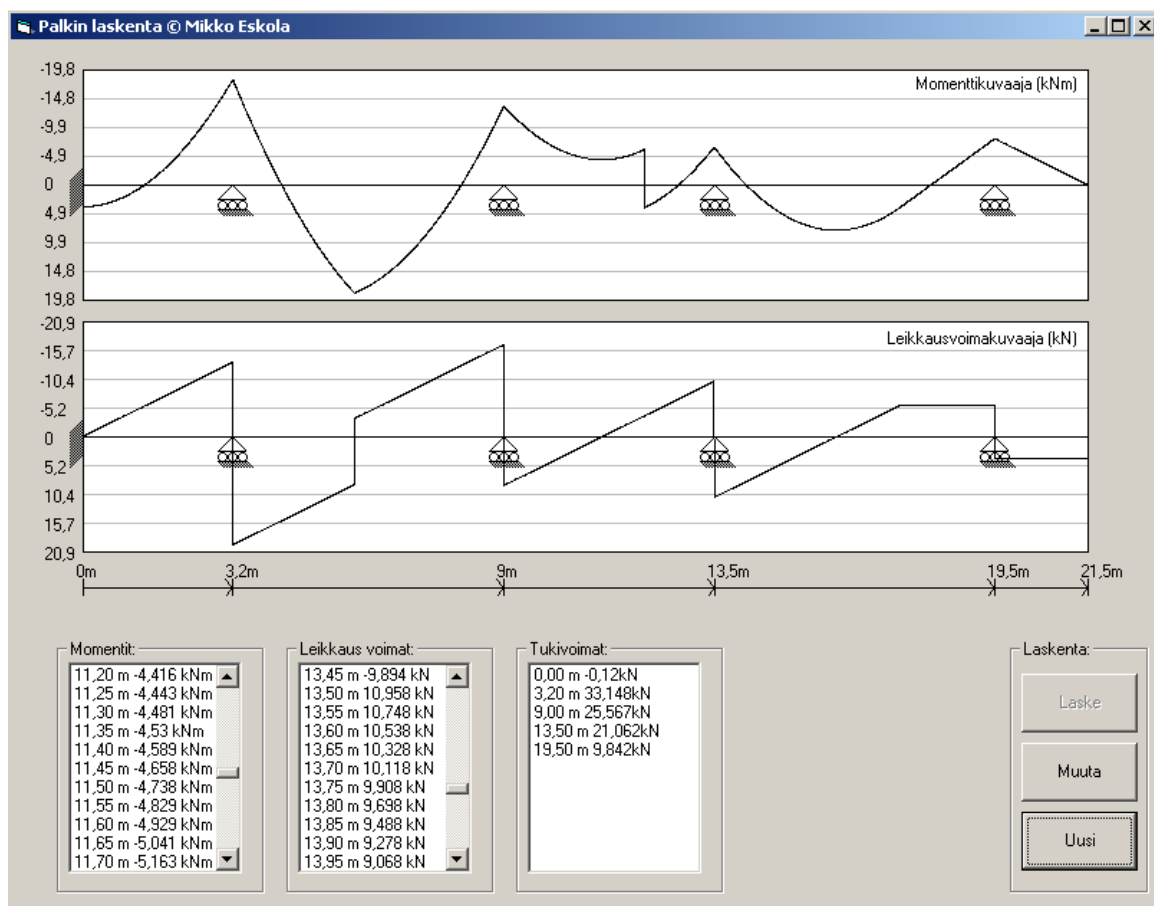
Tämän jälkeen rakennemalli tulisi näyttää kuvion 27 mukaiselta.



Kuvio 27: Rakennemalli käyttäjän syöttämien tietojen jälkeen

Tietojen syöttämisen jälkeen laskenta aloitetaan painamalla *Laske* -painiketta, joka sijaitsee ruudun oikeassa alareunassa.

Jos tiedot on syötetty esimerkin mukaisesti, tulossivu on kuvion 28 mukainen.



Kuvio 28: Esimerkkilaskusta saadut tulokset

Tulossivulta on mahdollista palata muuttamaan syötettyjä arvoja painamalla *Muuta* -painiketta. Kokonaan uuden palkin laskennan voi aloittaa painamalla *Uusi* -painiketta.

7 Ohjelman jatkokehittäminen

Ohjelmiin on helppo keksiä uusia ominaisuuksia ja uudistuksia. Seuraavaksi käsitellään joitakin ideoita ohjelman parantamiseksi ja kehittämiseksi edelleen.

7.1 Taipumanlaskenta

Rakenteiden mitoittamiseen kuuluu olennaisesti taipumanlaskenta. Rakentamismääräyskokoelmassa taipumille on annettu maksimirajat, joita ei saa ylittää. Taipumanlaskenta moniaukkoiselle palkille, jonka suhteellinen jäykkyys vaihtelee, vaatisi Mohrin- tai matriisimenetelmän soveltamista.

7.2 Mitoitus

Mitoituksella tarkoitetaan tietystä materiaalista tehdyn poikkileikkauksen kestävyysvarmistamista annetun materiaalin mitoitusnormin mukaisesti. Käytännössä ohjelman käyttämä kuvitteellinen palkki muutettaisiin vastaamaan haluttua materiaalista poikkileikkausta. Poikkileikkauksesta tarkastettaisiin esimerkiksi, kestääkö se palkille tulevia rasituksia, sekä muita normien määräyksiä, kuten ovatko taipumat sallituissa rajoissa.

7.3 Tietojen tallentaminen ja tulostaminen

Rakenteita mitoittaessa tulisi aina tallentaa itselleen dokumentit tehdyistä ja lasketuista rakenteista. Ohjelmaan on tarkoitus lisätä toiminto, jonka avulla käyttäjä pystyy tallentamaan palkin geometrian sekä annetut kuormat erilliseen tiedostoon.

Suunnitteilla on myös erillinen tulostussivu, jolle ohjelma keräisi momentti- ja leikkausvoimakuvaajat sekä maksimirasitukset. Myöhemmin sinne tulisivat myös tiedot taipumista ja mitoitetusta poikkileikkauksesta. Luonnollisesti tulostussivun voisi myös tulostaa tai tallentaa muotoihin *.doc tai *.pdf.

7.4 Verkkoversio

Internetin kasvu ja kehittyminen on avannut myös sovelluksille lukemattomia mahdollisuuksia. Mikä olisikaan helpompaa kuin käyttää ohjelmaa verkossa tavallisella internet-selaimella?

Verkossa toimiva versio toisi käyttäjälleen mm. seuraavia etuja:

- Ohjelmaa pystyisi käyttämään millä tahansa laitteella, jolla pääsee internetiin.
- Ohjelmaa pystyisi käyttämään mistä paikasta tahansa, jossa on internet-yhteys.
- Ohjelman päivitys tapahtuisi automaattisesti.
- Käyttäjä välttyisi ohjelman asentamiselta.

8 Yhteenveto

Ohjelman tekeminen on ollut haastava ja mielenkiintoinen projekti. Haasteita on ollut matkan varrella monia. Ensimmäiseksi piti opetella ohjelmointikielestä monia uusia asioita, eihän raportin laatija ollut aikaisemmin ohjelmoinut kuin jotain pientä MS-DOSin mukana tulleella QBasic-tulkilla. Onneksi internetistä löytyy sivustoja, joilla Visual Basicin saloja selvitetään. Kunniamaininnan ansaitsevat ohjelmointiputkan korkeatasoiset ohjelmointioppaat ja koodivinkit.

Opinnäytetyön tekeminen opetti myös paljon uusia asioita rakenteiden statiikasta: sellaisia asioita, joita ei rakenteiden mekaniikan kurssien vähäisestä opintopistemäärästä johtuen käydä lävitse. Yksi suhteellisen tuntemattomaksi jäänyt alue oli esimerkiksi pistemomentit.

Lopputuloksena tuli helppokäyttöinen ja selkeä ohjelma, jonka käyttö on nopeasti opettavissa. Ohjelmalla saadaan laskettua nopeasti palkin rasitukset ja nähdään millaiset momentti- ja leikkausvoimakuviot rasituksista syntyy. Tähän laadittuun perusrunkoon on tulevaisuudessa helppo lisätä taipumanlaskenta, mitoitus, tietojen tallentaminen ja tulostaminen, sekä mahdollisia muitakin parannuksia.

Lähteet

Iivonen, Aarre 2004. Luentomoniste: Rakenteiden mekaniikan jatkokurssi.

Jaakkola, Matti & Jaakkola, Pekka & Päijät-Hämeen koulutus konserni 2002. Visual Basic –harjoituksia. [online] <http://edu.phkk.fi/vb/>

Laaksonen, Antti 2002. Visual Basic –opas [online]
http://www.ohjelmointiputka.net/opas.php?tunnus=vbo_1

Loikkanen, Pentti 1975. Rakenteiden statiikka 2: Staattisesti epämääräiset sauvarakenteet.