Roterende schijven met niet constante dikte

Roterende schijven met niet constante dikte.

In de „Aantekeningen bij het college Voortgezette Sterkteleer" door ir C.M. Menken (pag 6 e.v.) wordt de diffenliaal vergelijking van dit probleem afgeleid. Tevens wordt aangegeven van welke functies $y(r)$ deze diff. vergl. gesloten oplossingen toelaat. Het numeriek integreren van de diff. vergl. is mogelijk doch zeker in dit stadium te moeilijk. Een voor de hand liggende methode om te komen tot een benaderingsoplossing wordt gegeven door Grammel (zie Technische Dynamik Biezeno/Grammel, deel 2, pag. 12 e.v.). Deze methode leidt tot voldoende nauwkeurige resultaten waarop hier niet verder wordt ingegaan.
Grammel splitst de schijf op in een aantal schijven met constante dikte. fig. 1.

Neem als breedte y_j het gemiddelde van $y(r)$ tussen r_j en r_{j-1}

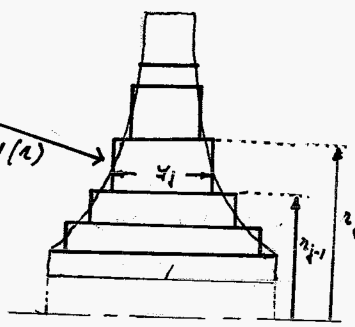


fig 1

Voor iedere schijf geldt (zie Menken pag. 5) als oplossing van de lineaire differentiaal vergelijking in de spanningen:

$$\sigma_r = A_1 + A_2/r^2 - \frac{3m+1}{8m}\frac{\gamma}{g}\omega^2 r^2 \qquad (1)$$

$$\sigma_t = A_1 - A_2/r^2 - \frac{m+3}{8m}\frac{\gamma}{g}\omega^2 r^2 \qquad (2)$$

waarin:

σ_r de radiale spanning in kg/cm^2

σ_t de tangentiële spanning in kg/cm^2

r de straal in cm
m de reciproque waarde van de dwarscontracte coëfficiënt
γ soortelijk gewicht in kg/cm³
g valversnelling in cm/min²
ω aantal omwentelingen per minuut
A_1 en A_2 integratie constanten

Door in te voeren $x = 1/r^2$ (3)

$$S = \sigma_r + \alpha \omega^2 r^2 \qquad (4)$$

$$t = \sigma_t + \beta \omega^2 r^2 \qquad (5)$$

waarin $\alpha = \frac{3m+1}{8m} \frac{\gamma}{g}$ en $\beta = \frac{m+3}{8m} \frac{\gamma}{g}$ (6)

krijgen we eenvoudige uitdrukkingen voor (1) en (2)

$$S = A_1 + A_2 x \qquad (7)$$

$$t = A_2 - A_2 x \qquad (8)$$

Zijn nu de spanningen σ_{rj} en σ_{tj} aan de buitenrand van een tussenschijf j met buitenstraal r_j bekend, dus ook s_j en t_j, dan kunnen we met (3), (7) en (8) de constanten A_1 en A_2 voor die schijf berekenen.
Immers $s_j = A_1 + A_2 x_j$ en $t_j = A_1 - A_2 x_j$
Hieruit volgt s_{j-1} en t_{j-1} voor de binnenzijde van de schijf met straal r_{j-1} : $s_{j-1} = A_1 + A_2 x_{j-1}$ en $t_{j-1} = A_1 - A_2 x_{j-1}$ (9)
waaruit met (4) en (5) de spanningen σ_{rj-1} en σ_{tj-1} volgen
We noteren dit als volgt (zie ook fig 2):

$$s_j - t_j = 2 A_2 x_j \quad \Rightarrow \quad A_2 = \frac{s_j - t_j}{2 x_j}$$

$$s_j + t_j = 2 A_1 \quad \Rightarrow \quad A_1 = \frac{s_j + t_j}{2}$$

sub in (9)

$$s_{j-1} = \frac{s_j + t_j}{2} + \frac{s_j - t_j}{2 x_j} (x_{j-1})$$

$$= \frac{s_j x_j + t_j x_j + s_j x_{j-1} - t_j x_{j-1}}{2 x_j}$$

$$s_{j-1} = s_j + \frac{-s_j x_j + t_j x_j + s_j x_{j-1} - t_j x_{j-1}}{2 x_j}$$

$$s_{j-1} = s_j + \frac{(x_{j-1} - x_j)}{2 x_j}(s_j - t_j)$$

Stellen we $q = \frac{x_{j-1} - x_j}{2 x_j}(s_j - t_j)$ (9)'

dan krijgen we $s_{j-1} = s_j + q$ (10)
en eveneens $t_{j-1} = t_j - q$ (11)

fig 2

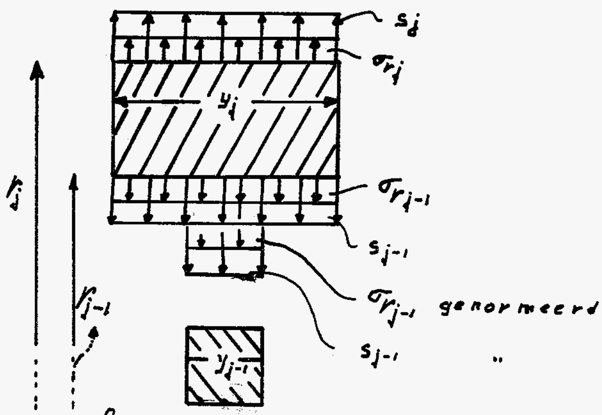
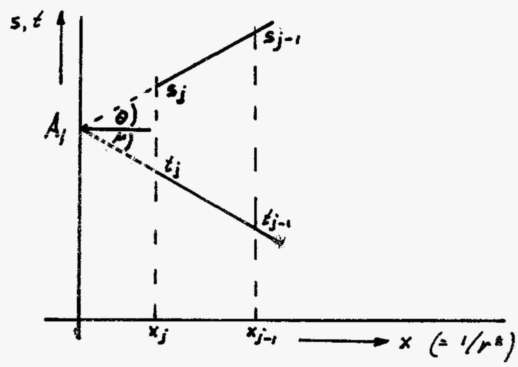


fig 2.

Grafisch is dit als volgt te zien

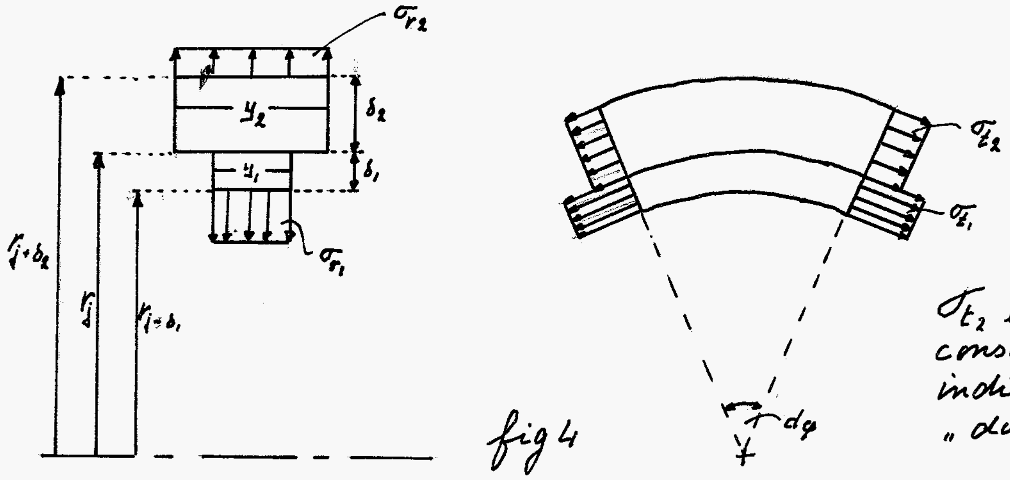


∠ θ wordt bepaald door A_2
∠ μ wordt bepaald door $-A_2$
immers $s = A_1 + A_2 x$
$t = A_1 - A_2 x$
$s - t = 2 A_2 x$

fig 3.

Uit fig 2 volgt dat de berekende spanningen $\sigma_{r_{j-1}}$ en s_{j-1} zo ook $\sigma_{t_{j-1}}$ en t_{j-1} fictieve spanningen zijn die nog genormeerd moeten worden op de breedte van schijf $j-1$.

De sprong in de radiale spanning bij de overgang van schijf j op schijf j-1 bepalen we met behulp van het krachtenevenwicht in radiale richting van een schijfsegment:



σ_{t_2} en σ_{t_1} mogen we constant veronderstellen indien de schijven „dun" zijn.

fig 4

$$\sigma_2 \cdot y_2 (r+\delta_2)\, d\varphi + \alpha\,\omega^2 (r+\tfrac{1}{2}\delta_2)^2\, y_2\, (r+\tfrac{1}{2}\delta)\, d\varphi\, \delta_2 +$$
$$-\sigma_{r_1} \cdot y_1 (r-\delta_1)\, d\varphi + \alpha\,\omega^2 (r-\tfrac{1}{2}\delta_1)^2\, y_1\, (r-\tfrac{1}{2}\delta)\, d\varphi\, \delta_1 +$$
$$-\sigma_{t_2}\, y_2\, \delta_2\, d\varphi - \sigma_{t_1} \cdot y_1\, \delta_1\, d\varphi = 0$$

Wanneer δ_1 en $\delta_2 \to 0$ (in de overgang!)

$$\sigma_{r_2} \cdot y_2\, r - \sigma_{r_1}\, y_1\, r = 0$$
$$\sigma_{r_2}\, y_2 - \sigma_{r_1}\, y_1 = 0 \qquad (12)$$

Bij de overgang geldt sprong in σ_r = sprong in s

immers
$$\begin{aligned} s + \Delta s &= \sigma_r + \Delta\sigma_r + \alpha\,\omega^2 r^2 \\ s &= \sigma_r + \alpha\,\omega^2 r^2 \quad - \\ \hline \Delta s &= \Delta\sigma_r \end{aligned}$$

evenzo is $\Delta t = \Delta\sigma_t$

De op schijf j-1 genormeerde spanning noemen we $\sigma_r^* \to s^*$

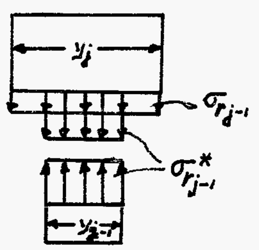


fig 5

Uit (12) volgt: $\sigma_{r_{j-1}}\, y_2 = \sigma^*_{r_{j-1}}\, y_{j-1}$

$$-\sigma_{r_{j-1}} + \sigma^*_{r_{j-1}} = \Delta\sigma_{r_{j-1}} = \Delta s_{j-1} = -\sigma_{r_{j-1}} + \sigma_{r_{j-1}} \frac{y_j}{y_{j-1}} \Rightarrow \Delta s_{j-1} = \frac{-y_{j-1} - y_j}{y_{j-1}} \cdot \sigma_{r_{j-1}} \qquad (13)$$

De sprong in de tangentiële spanning bij de overgang bepalen we uit de contabiliteitseis m.a.w. uit de continuïteit van de radiale verplaatsing u.

$$u = \frac{r}{E}\left(\sigma_t - \frac{1}{m}\sigma_r\right)$$

$$u^* = \frac{r}{E}\left(\sigma_t^* - \frac{1}{m}\sigma_r^*\right) \quad -$$

$$\Delta u = 0 = \frac{r_{j-1}}{E}\left(\Delta\sigma_t - \frac{1}{m}\Delta\sigma_r\right) \Rightarrow \Delta\sigma_{t_{j-1}} = \frac{\Delta\sigma_{r_{j-1}}}{m}$$

dus $$\Delta t_{j-1} = \frac{\Delta s_{j-1}}{m} \qquad (14)$$

$$\left.\begin{aligned} s_{j-1}^* &= s_{j-1} + \Delta s_{j-1} \\ \text{en} \quad t_{j-1}^* &= t_{j-1} + \Delta t_{j-1} \end{aligned}\right\} \qquad (15)$$

Voor de buitenrand van schijf j was $\Delta s_j = \Delta t_j = 0$
zodat ook hier gold: $$\left.\begin{aligned} s_j^* &= s_j + \Delta s_{j-1} \\ t_j^* &= t_j + \Delta t_{j-1} \end{aligned}\right\} \qquad (15)'$$

Willen we uniform te werk gaan dan voeren vergl (10) en (11) over in:
$$s_{j-1} = s_j^* + q \qquad (10)'$$
$$t_{j-1} = t_j^* - q \qquad (11)'$$

Geven we de binnen~~zij~~deste schijf rangnummer 2 met binnenstraal r_1 en breedte y_2 en laten we de vergelijkingen (15)' (9)' (10)' en (11)' $(n-1)$ maal los op de steeds nieuw gevonden waarden van de verschillende parameters, dan hebben we hiermee een procedure ontwikkeld, die bij gegeven randspanningen σ_{r_n} en σ_{t_n} aan de buitenkant, berekent alle spanningen σ_{r_j} en σ_{t_j} voor $1 \leq j \leq n$.

In bijgevoegd computerprogramma is deze procedure opgenomen onder de naam procedure GRAMMEL

Algoritmisch betekent een „procedure" een stukje programma, dat onder een bepaalde naam (hier GRAMMEL) als zodanig in de kop van een hoofdprogramma staat vermeld en dat later naar wens aangeroepen kan worden waarna de procedure uitgevoerd wordt. (zie RC informatie)

In het algemeen zijn niet de randvoorwaarden σ_{rn} en σ_{tn} gegeven doch b.v. σ_{rn} en σ_{r1}
In dat geval nemen we een willekeurige waarde σ_{tn}' aan en berekenen met behulp van $\sigma_{rn}' = \sigma_{rn}$ en σ_{tn}' de spanningen σ_{r1}' en σ_{t1}' aan de binnenzijde en de tussenliggende spanningen. Hiermede hebben we een spanningstoestand I gevormd die wel wat betreft σ_{rn} aan de gegeven randvoorwaarde voldoet doch verder in het algemeen nergens.
We creëren nu een tweede spanningstoestand II met willekeurige randvoorwaarden σ_{tn}'' met $\omega = 0$ en $\sigma_{rn}'' = 0$ waaruit met de procedure volgt een σ_{r1}'' en σ_{t1}''

Vermenigvuldigen we nu spanningstoestand II met een bepaald getal k en tellen deze op bij spanningstoestand I dan krijgen we een spanningstoestand die én aan de buiten-randvoorwaarde $\sigma_{rn} = \sigma_{rn}' + k.0$ én aan de binnenrandvoorwaarde $\sigma_{r1} = \sigma_{r1}' + k\sigma_{rg1}''$ (16) voldoet.
i.p.v. σ_r nemen we de gem. spanning $\sigma_r + 1/2\, \Delta s$
zodat $\sigma_{r1} = \sigma_{rg1}' + k\, \sigma_{rg1}''$
Deze laatste spanningstoestand nemen we aan als benadering voor de werkelijke toestand.
Dat superpositie geoorloofd is, volgt uit het lineair-zijn van de beschrijvende tweede orde differentiaalvergelijking.

Men kan derhalve toestand I opvatten als oplossing van het particuliere deel van de differentiaalvergelijking waarin een randvoorwaarde reeds verwerkt zit, en toestand II als oplossing van het homogene gedeelte van de dif. vergl. waarbij de integratie constante k volgt uit (16); σ_{r1} is immers de tweede gegeven randconditie

Uit de praktijk weten we dat het s - x verband van de volgende aard is:

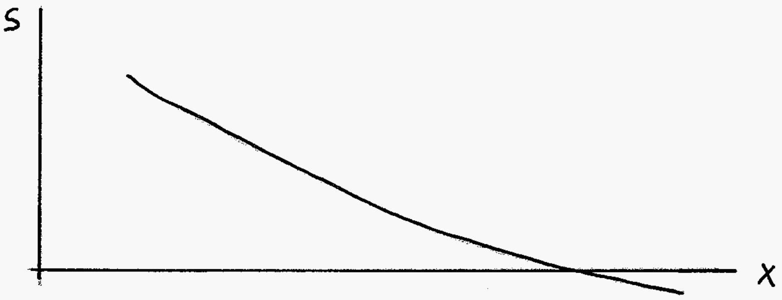


fig 6

In de procedure Grammel wordt deze kromme benaderd met rechte stukken tussen de x_j's

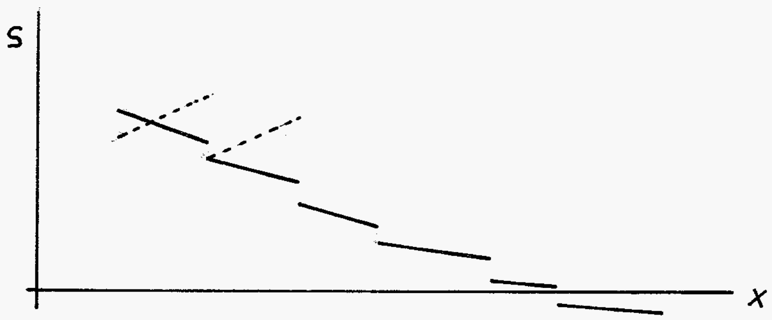


fig. 7

waaruit volgt dat $\frac{ds}{dx} \leq 0$ met andere woorden $A_2 \leq 0$

(immers het zou onjuist zijn de kromme te benaderen met de gestippelde rechten; $A_2 > 0$)

Met $s - t = 2Ax$, $x > 0 \Rightarrow s - t \leq 0 \Rightarrow t \geqslant s$
Neem dan als startwaarde $t = s$

$$\text{m.a.w. } \sigma_t + \beta \omega^2 r^2 = \sigma_r + \alpha \omega^2 r^2$$

$$\sigma_t = \sigma_r + (\alpha - \beta)\omega^2 r^2 \qquad (17)$$

In het programma zijn acht combinaties met gegeven randvoorwaarden opgenomen.

Comb 1 : Gegeven randvoorwaarden σ_{rn} en u_n. Hiermee is ook σ_{tn} gegeven immers: $\varepsilon_t = \frac{u}{r}$

$$u = r \cdot \frac{1}{E}(\sigma_t - \nu\sigma_r) \qquad (18)$$

zodat $$\sigma_t = \frac{u \cdot E}{r} + \nu\sigma_r \qquad (19)$$

met $r = r_n$ en $\sigma_r = \sigma_{rn}$ volgt hieruit σ_{tn}. We hoeven nu slechts één keer Grammel uit te voeren.

Comb 2 — Gegeven randconditie σ_{r_1} en u_1
Met $r = r_1$ en (19) kunnen we σ_{t_1} berekenen. Het maakt voor het gebruik van de procedure Grammel geen verschil of we van buiten naar binnen werken of omgekeerd. Het enige wat we slechts hoeven te doen is de stralen r_j en de heedten y_j in omgekeerde volgorde te laten inlezen. Indien we voor de boolean variabele comb de waarde 2 opgeven via de getallenband, wordt dit in het programma voor ons gedaan zodat we ook in dit geval de r_j's en de y_j's in de normale volgorde moeten opgeven.
De procedure behoeft ook nu slechts één keer te worden uitgevoerd.

Comb 3 — Gegeven u_1 en σ_{rn}
Kies m.b.v. (17) $\sigma_{tn} \Rightarrow$ GRAMMEL σ_{r_1}' en $\sigma_{t_1}' \Rightarrow (18) \Rightarrow u_1'$
Toestand II Neem $\omega = 0$ en $\sigma_{rn} = 0$ Kies een $\sigma_{tn}'' \neq 0$
$\Rightarrow$ GRAMMEL σ_{r_1}'' en $\sigma_{t_1}'' \Rightarrow (18) \Rightarrow u_1''$
k volgt weer uit $u = u_1' + k\, u_1''$
Vermenigvuldig toestand II met k en tel deze op bij spanningstoestand I

Comb 4 — Voorgeschreven u_n en σ_{r_1}
Kies m.b.v (17) $\sigma_{t_1}' \Rightarrow$ GRAMMEL σ_{rn}' en $\sigma_{tn}' \Rightarrow u_n'$
Kies $\omega = 0$ en $\sigma_{r_1}'' = 0$ en een willekeurige $\sigma_{t_1}'' \neq 0$
Grammel levert weer σ_{rn}'' en $\sigma_{tn}'' \Rightarrow u_n''$
k wordt bepaald door $u_n = u_n' + k\, u_n''$. Het gevraagde spanningsverloop volgt weer uit superpositie van I en k II

Comb. 5 Gegeven U_1 en U_n

I kies een willekeurige waarde $\sigma_{tn}' \Rightarrow (19): \sigma_{tn}$, Grammel: σ_{r_1}' en $\sigma_{t_1}' \Rightarrow U_1'$

II kies $U_n'' = 0$, $\omega = 0$, $\sigma_{rn}'' \Rightarrow \sigma_{tn}'' \Rightarrow$ Grammel: σ_{r_1}'', $\sigma_{t_1}'' \Rightarrow U_1''$
k wordt bepaald door $U_1 = U_1' + k\,U_1''$
gevraagde spanningstoestand = I + kII

Comb. 6 Gegeven σ_{r_1} en σ_{r_n}

I Kies m.b.v. (17) $\sigma_{tn}' \Rightarrow$ Grammel $\Rightarrow \sigma_{r_1}'$ en σ_{t_1}'

II Neem $\omega = 0$ en $\sigma_{rn}'' = 0$ kies σ_{tn}'' $(\neq 0) \Rightarrow \sigma_{r_1}'$ en σ_{t_1}''
k volgt uit $\sigma_{r_1} = \sigma_{r_1}' + k\,\sigma_{r_1}''$
Vermenigvuldig toestand II met k en tel op bij I

Comb. 7 Schijf zonder gat. U_n voorgeschreven

Voor de binnenschijf wil dit zeggen $A_2 = 0$ of $\sigma_r = \sigma_t \Rightarrow$
$\Rightarrow s = t$. De procedure wordt van binnen naar buiten uitgevoerd. ($r[n] = 0$).
Staat de schijf stil, dan is per schijf σ_r en σ_t constant.
Bij rotatie nemen de spanningen naar binnengaand toe t.g.v. de termen $\alpha\omega^2 r^2$ resp. $\beta\omega^2 r^2$.
We kiezen bij $r[n] = 0$ spanningen σ_{rn}' en $\sigma_{tn}' \Rightarrow$
$\Rightarrow$ Grammel: σ_{r_1}' en σ_{t_1}'. Bij de overgangen middelen we de spanningen: $\bar{\sigma}_{r_i}'$ en $\bar{\sigma}_{t_i}'$.
In het algemeen zijn deze spanningen niet gelijk aan die welke moeten voldoen aan: $U = \frac{r}{E}(\sigma_t - {}^1\!/\!_m\,\sigma_r)$.

We vermeerderen daartoe de gevonden spanningen met een waarde ka.
$U_1 = \frac{r[1]}{E}\left\{(\bar{\sigma}_{t_1}' + ka) - {}^1\!/\!_m(\bar{\sigma}_{r_1}' + ka)\right\}$

Hieruit volgt de waarde van ka zodat ook aan deze randvoorwaarde is voldaan.

Comb. 8 Schijf zonder gat. σ_{rn} gegeven.

Evenals onder 7. De waarde van ka volgt uit
σ_{r_1} (gegeven) $= \bar{\sigma}_{r_1}' + ka$

Om numeriek niet in moeilijkheden te komen, komt in de procedure Grammel x[j] niet voor. We voeren daarin: $d = r[j] - r[j-1]$ en berekenen hiermede q. Iets dergelijks geldt voor de breedte y[j].

In algol kunnen we de Griekse letters α en β evenals de accenten en sterren zoals boven gebezigd niet gebruiken. We hebben daarvoor ingevoerd:

$\alpha = a$

$\beta = b$

$\left.\begin{array}{l}\sigma_r = p_r \\ \sigma_t = p_t\end{array}\right\}$ het cijfer 1 resp 2 slaat op eerste resp tweede ronde waarin we de schijven 1 - n doorlopen

$s^* = ss$, $\Delta s = ds$

$t^* = ts$, $\Delta t = dt$

$k = k.a$

$\omega = v$ (de w in het programma stelt de hoeksnelheid $2\pi n$ voor)

Men dient zich steeds te realiseren wat de „indices" 1 en n betekenen, vooral bij het omgekeerd werken met de procedure Grammel. Het tekenen van een plaatje geeft voldoende duidelijkheid.

De ingebouwde procedure Graf tekent σ_r en σ_t als functie van r. Het is niet de bedoeling deze procedure te bestuderen.

17 - 2 - '71 L. Slenter

```
Lalgol 05064788 SLENTER,0,1
begin comment Berekening van roterende schijven met niet constante doorsnede
      volgens de methode van grammel met verschillende voorgeschreven
      randvoorwaarden
 Volgorde van inlezen
      KK      integer, aantal malen, dat het programma in zijn geheel uitgevoerd moet worden (aantal schijven)
      n       integer ,aantal sneden, aan binnenrand n = 1, buitenrand n =n (aantal stralen)
      comb    integer,  afhankelyk van gegeven randvoorwaarden
              Indien voorgeschreven
              met gat spanning buitenrand - verplaatsing buitenrand comb = 1
                      spanning binnenrand - verplaatsing binnenrand comb = 2
                      spanning buitenrand - verplaatsing binnenrand comb = 3
                      spanning binnenrand - verplaatsing buitenrand comb = 4
                      verplaatsing binnenrand - verplaatsing buitenrand comb = 5
                      spanning binnenrand - spanning buitenrand comb = 6
              zonder gat verplaatsing buitenrand comb = 7
                         spanning buitenrand comb = 8
      E       real, elasticiteits-modulus   [kgf/cm ∧ 2]
      r[1|n]  array, straal van de sneden   r[1] = straal binnenrand    r[n] = straal buitenrand    [cm] beginnen met r[n]
      y[2|n]  array, breedte van de schyven  y[2] = breedte binnenste schyf  y[n] = breedte buitenste schyf [cm] BEGINNEN MET Y[n]
      m       1/poisson-constante
      gamma   real, soortelyk gewicht   [kgf/cm ∧3]
      g       real, valversnelling   [cm/min ∧ 2]
      v       real, aantal omwent./min
      Pr1     real, radiale spanning  [kgf/cm ∧ 2] voor comb 1,  3, 6    gegeven radiale buitenspanning
                                                   voor comb 2 ,4      gegeven radiale biinenspanning
                                                   voor comb 5         gekozen radiale buitenspanning
                                                   voor comb 7,8       gekozen binnenspanning
      combinatie 1
      Un              real, voorgeschreven verplaatsing aan de buitenrand   [cm]
      combinatie 2
      U               real, voorgeschreven verplaatsing binnenzyde
      combinatie 3
      Ph1             real, tangentiele spanning [kgf/cm ∧2] aan de buitenrand kiezen
      Ph2             real, tang.      spann.              aan de buitenrand kiezen tweede ronde
      Uo              real, verplaatsing [cm] aan de binnenrand voorgeschreven
      combinatie 4
      U               real, voorgeschreven verplaatsing buitenrand [cm]
      Ph1             real, e kiezen tang spanning [kgf/cm ∧ 2] aan de binnenzijde eerste ronde
      Ph2             reil, te kiezen tang spanning [ ,,    ,,  ] aan de binnenzijde tweede ronde
      combinatie 5
      Uo              voorgeschreven verplaatsing binnenrand  [cm]
      Un              voorgeschreven verplaatsing buitenrand  [cm]
      combinatie 6
```
11

```
    Ph1            real, te kiezen tang. spanning aan de buitenrand  [kgf/cm ⏶2]
    Ph2            real, te kiezen tang. spanning aan de buitenrand 2de ronde
    Pr1            real, gegeven radiale binnenspanning
    combinatie 7
    U              real, voorgeschreven verplaatsing aan de buitenrand  [cm]
    combinatie 8
    Pr1            real, voorgeschreven rad. spanning aan de buitenrand  [kgf/ cm ⏶ 2];

integer K, KK; KK:= read; for K:= 1 step 1 until KK do
 begin if K > 1then NEWPAGE; PRINTTEXT (⪡ KK = ⪢); FIXT( 2,0,KK);

begin
    procedure GRAMMEL (m, gamma, g, w, n, r, y, Pr, Ph, ds, dt);
    value m, gamma, g, w, n; real m, gamma, g, w; integer n; array r, y, Pr, Ph, ds, dt;
    begin real a,d, b, p, s, t, ss, ts, q; integer j, i;
        a:= ((3 × m + 1)/(8 × m)) × gamma/g;  NLCR; PRINTTEXT ( ⪡ aw⏶2 = ⪢ ) ; FLOT ( 5,3,a×w⏶2 ) ; NLCR ;
        b:= ((m + 3)/(8 × m)) × gamma/g; PRINTTEXT ( ⪡ bw ⏶ 2 = ⪢ ) ; FLOT ( 5,3,b×w ⏶2 ) ; NLCR ;
        s:= Pr[n] + a × (w ⏶ 2) × r[n] ⏶ 2;
        t:= Ph[n] + b × (w ⏶ 2) × r[n] ⏶ 2;
        CARRIAGE(3); PRINTTEXT(⪡  n         sigma-r            delta-s            sigma-h            delta-t⪢); NLCR;
        for j:= n step - 1 until 2 do
        begin NLCR; FIXT (2, 0, j-1);
            ss:= s + ds[j];
            ts:= t + dt[j];
            d := r[j] - r[j-1];
            q := d/r[j-1] × (1 + 1/2×d/r [j-1] )×(ss - ts);
            s:= ss + q;
            Pr[j - 1]:= s - a × r[j - 1] ⏶ 2 × w ⏶ 2; SPACE(5); FLOT(5,3,Pr[j - 1]);
            p:= y[j] - y[j-1];
            ds[j-1]:= p/y[j-1] × Pr[j-1]; SPACE (5); FLOT(5,3,ds[j-1]);
            t:= ts - q;
            Ph[j - 1]:= t - b × r[j - 1] ⏶ 2 × w ⏶ 2; SPACE(5); FLOT(5,3,Ph[j - 1]);
            dt[j - 1]:= ds[j - 1]/m; SPACE(5); FLOT(5,3,dt[j - 1]);
        end
    end procedure grammel;

    procedure UITVOER(ka, Pr1, ds1, Ph1, dt1);
    real ka; array Pr1, ds1, Ph1, dt1;
    begin integer l;
        CARRIAGE(3);
        PRINTTEXT(⪡  n          sigma-Prg            sigma-Pr            sigma-Phg            sigma-Ph⪢);
        for l := n step - 1 until 1 do
        begin NLCR; FIXT(2, 0, l);
            Pr1[l] := Pr1[l] + 1/2 × ds1[l]; SPACE(5); FLOT(5, 3, Pr1[l]);
            Pr1[l] := Pr1[l] × ka          ; SPACE(5); FLOT(5, 3, Pr1[l]);
```

```
            Ph1[l] := Ph1[l] + 1/2 × dt1[l]; SPACE(5); FLOT(5, 3, Ph1[l]);
            Ph1[l] := Ph1[l] × ka          ; SPACE(5); FLOT(5, 3, Ph1[l]);
        end
    end procedure UITVOER;

    procedure UITVOER2 (ka, Pr1, Ph1, Pr2, Ph2, ds1, ds2, dt1, dt2 );
    real ka ; array Pr1, Ph1, Pr2, Ph2, ds1, ds2, dt1, dt2;
    begin integer l;
        CARRIAGE (3);
        PRINTTEXT (⊀  n          sigma-rg1          sigma-rg2          sigma-r          sigma-hg1          sigma-hg2
  sigma-h ⊁);NLCR;
        for l:= n step -1 until 1 do
        begin NLCR; FIXT (2,0,1);
        Pr1[l] :=Pr1[l] + 1/2 × ds1[l]; SPACE (5); FLOT(5,3,Pr1[l]);
        Pr2[l] :=Pr2[l] + 1/2 × ds2[l]; SPACE (5); FLOT(5,3,Pr2[l]);
        Pr1[l] :=Pr1[l] + ka× Pr2[l]; SPACE (5); FLOT(5,3,Pr1[l]);
        Ph1[l] :=Ph1[l] + 1/2×dt1[l]; SPACE (5); FLOT(5,3,Ph1[l]);
        Ph2[l] :=Ph2[l] + 1/2×dt2[l]; SPACE (5); FLOT(5,3,Ph2[l]);
        Ph1[l] :=Ph1[l] + ka×Ph2[l]; SPACE  (5); FLOT(5,3,Ph1[l]);
        end
    end procedure UITVOER 2;


    procedure GRAF(N, x, fx, l, nintx, h, ninty, scale, ii, first, sort, curve, lin);
value N, l, nintx, h, ninty, ii; integer l, nintx, h, ninty, ii, N; array x, fx, scale; boolean first, sort, curve, lin;
begin comment GRAFiek, voor nadere gegevens zie :Toelichting op WB-procedures;
    integer i, n1, n2;
    own real XMAX, XMIN, DX, Q, YMAX, YMIN, DY, R;
    boolean nulx, nuly, S;

    procedure SORT(N,x,fx);
    value N; integer N; array x, fx;
    begin integer m, k; real s; boolean wissel;
        for m := 2 step 1 until N do
        begin wissel:= true; k := -1; for k := k + 1 while k < m - 1 ∧ wissel do
            begin if x[m-k]<x[m-k-1] then begin s := x[m - k]; x[m - k] := x[m - k - 1]; x[m - k - 1] := s;
                                                s := fx[m - k]; fx[m - k] := fx[m - k - 1]; fx[m - k - 1] := s
                                          end
                                     else wissel:= false
            end
        end
    end SORT;

    S:=true;
    if first then begin if sort then SORT(N,x,fx); PLOTFRAME(1, 1, 2, 2, 1, 1); PLOT(-1, 0, -4);
                        if scale[1]≠scale[2] then SCALE(scale[i], i, 2, nintx, 0, XMIN, XMAX, DX)
```

```
                                 else SCALE(x[i], i, N, nintx, 0, XMIN, XMAX, DX);
            if scale[3]≠scale[4] then SCALE(scale[i+2], i, 2, ninty, 0, YMIN, YMAX, DY)
                                 else SCALE(fx[i], i, N, ninty, 0, YMIN, YMAX, DY);
            mulx:=sign(XMAX)×sign(XMIN)=-1; muly:=sign(YMAX)×sign(YMIN)=-1;
            Q := (XMAX - XMIN)/(100 × l); R := (YMAX - YMIN)/(100 × h);
            PLOTFRAME(XMIN - Q × 500, YMIN - R × 100, XMAX + Q × 1300, YMAX + R × 50, l × 100 + 1800, h × 100 +
150);
            PLOTAXIS2(XMIN, XMAX, DX, true, if muly then 0 else YMIN);
            PLOTAXIS2(YMIN, YMAX, DY, false, if mulx then 0 else XMIN);
            PLOTTEXT(XMAX + Q × 50, if muly then 0 else YMIN, 0, 30, 0, true, 10, ⊰onafh. var. ⊱);
            PLOTTEXT(if mulx then 0 else XMIN, YMAX + R × 10, 0, 30, 0, true, 10, ⊰afh. var. ⊱);
            n2:=N; scale[1]:=XMAX+500×Q; scale[2]:=YMAX; scale[3]:=Q; scale[4]:=R;
        end

    else begin  n1 := 0;
            for i := 1 step 1 until N do
            begin if fx[i] > YMIN - 80 × R ∧ fx[i] < YMAX + 30 × R then
                  begin n1 := n1 + 1; fx[n1] := fx[i]; x[n1] := x[i] end
            end; S := n1 = N;
            if sort then SORT(n1, x, fx); n2 := 0;
            for i := 1 step 1 until n1 do
                  begin if x[i] > XMIN - 480 × Q ∧ x[i] < XMAX + 500 × Q then
                  begin n2 := n2 + 1; x[n2] := x[i]; fx[n2] := fx[i] end
            end; S := S ∧ n2 = n1
        end;

  NLCR; PRINTTEXT(⊰    XMIN              XMAX              DX              YMIN              YMAX              DY⊱);
  NLCR; FLOT(5,3,XMIN); SPACE(5); FLOT(5,3,XMAX); SPACE(5); FLOT(5,3,DX); SPACE(5);
        FLOT(5,3,YMIN); SPACE(5); FLOT(5,3,YMAX); SPACE(5); FLOT(5,3,DY);
  NLCR; if ¬ S then PRINTTEXT(⊰niet⊱); PRINTTEXT(⊰ alle gegevens worden verwerkt⊱);
  if ¬ curve ∧ ¬ lin then for i := 1 step 1 until n2 do PLOTTEXT(x[i], fx[i], 0, 28, 0, true, -ii, ⊰⊱);
  if curve ∧ ¬ lin then begin PLOTCURVE(0, 0, 1); PLOTTEXT(x[1], fx[1], 0, 28, 0, true, -ii, ⊰⊱);
                        for i := 1 step 1 until n2 do PLOTCURVE(x[i], fx[i], 2); PLOTCURVE(0, 0, 3);
                        PLOTTEXT(x[n2], fx[n2], 0, 28, 0, true, -ii, ⊰⊱)
                    end;
  if curve ∧ lin then begin PLOTTEXT(x[1], fx[1], 0, 28, 0, true, -ii, ⊰⊱);
                      for i := 1 step 1 until n2 do PLOT(x[i], fx[i], 1);
                      PLOTTEXT(x[n2], fx[n2], 0, 28, 0, true, -ii, ⊰⊱)
                  end
end GRAF;
```

14

```
procedure GRAFIEK (n, Pr1, Ph1, r, y);
value n ; array Pr1, Ph1, r, y; integer n;
begin integer i, j, l, h, nintx, ninty, ii;
  array x, fx [1 : 2 × n], scale[1 : 4];
  boolean first, sort, curve, lin;
  real maxabsfx, factor, ymax;
  maxabsfx:= abs(Pr1[1]);
  for i:= 1 step 1 until n do
  begin if abs(Pr1[i]) > maxabsfx then maxabsfx:= abs(Pr1[i]);
        if abs(Ph1[i]) > maxabsfx then maxabsfx:= abs(Ph1[i]);
  end;
  for j:= 1 step 1 until n do y[j]:= 1/2 × y[j];
  for i:= 1 step 1 until n do
  begin j:= (i - 1) + i; x[j]:= x[j + 1]:= r[i] end;
  ymax:= y[1];
  for i:= 2 step 1 until n do if y[i] > ymax then ymax:= y[i];

  factor:= 5 × ymax/maxabsfx;
  for i:= 1 step 1 until n do y[i]:= y[i]/factor;
  y[1]:= 0;
  for i:= 1 step 1 until n - 1 do
  begin j:= (i - 1) + i;
        fx[j]:= y[i]; fx[j + 1]:= y[i + 1];
  end;
  fx[2×n - 1]:= fx[2×n - 2]; fx[2×n]:=0;

  CARRIAGE(3); PRINTTEXT(≮  n         U(r) ≯); for i:= 1 step 1 until n do begin NLCR; FIXT(2,0,i);
                                            factor:= r[i]/EX(Ph1[i] - 1/m × Pr1[i]); SPACE(5); FLOT(5,3,factor); end;
  NLCR; PRINTTEXT (≮ r= ≯); SPACE(3); for i:=1 step 1 until 2×n do begin FIXT(2,0,x[i]); SPACE(3); end;
  NLCR; PRINTTEXT (≮ y= ≯); SPACE(3); for i:=1 step 1 until 2×n do begin FLOT(4,2,fx[i]); SPACE(3)  end;

 ymax:= r[1]; for i:= 1 step 1 until n do if r[i] > ymax then ymax:= r[i];
  scale[1]:= 0; scale[2]:= ymax;
  scale[3]:= -maxabsfx; scale[4]:= maxabsfx;
  GRAF(2 × n, x, fx, 10, 10, 20, 20, scale,2 , true, false, true, true);
  PLOTTEXT(scale[1], scale[2], 0, 30, 0, true, 10, ≮ster=radiale spann≯);
  PLOTTEXT(scale[1], scale[2] - 1 × 60 × scale[4], 0, 30, 0, true, 10,≮vierkant=tang spann≯);
  GRAF(n, r, Pr1, 10, 10, 20, 20, scale, 5, false, false, true, false);
  GRAF(n, r, Ph1, 10, 10, 20, 20, scale, 9, false, false, true, false)
end grafiek;
```

```
library SCALE, PLOTTEN, PLOTAXIS, PLOTAXIS2;
integer i, n, k ,comb; real m, gamma, g, w, v, pi, E;
n := read; comb := read; E := read;
begin array r, Pr1, Ph1, y, ds1, dt1[1:n];
      integer i; NLCR;
      for i := n step - 1 until 1 do r[i] := read;
      for k := n step - 1 until 2 do y[k] := read; y[1] := y[2];
      m := read; gamma := read; g := read;
      ds1[n] := dt1[n] := 0; pi := 4 × arctan(1); v := read; w := 2 × pi × v;
      Pr1[n] := read;

      if comb = 1 then begin NLCR; PRINTTEXT(⊀combinatie 1, voorgeschreven Prn en Un⊁); NLCR;
                       begin  real ka, Un; Un := read; ka:= 1;
                             Ph1[n] := 1/m × Pr1[n] + Un × E/r[n];
                             GRAMMEL(m, gamma, g, w, n, r, y, Pr1, Ph1, ds1, dt1);
                             UITVOER(ka, Pr1, ds1, Ph1, dt1);
                             GRAFIEK(n, Pr1, Ph1, r, y)
                       end
                       end;

      if comb = 2 then begin NLCR; PRINTTEXT(⊀combinatie 2, voorgeschreven Pro en Uo⊁); NLCR;
                             begin integer i; real ka, U; array R ,Y[1:n];
                                   for i := 0 step 1 until n - 1 do
                                    R[i + 1] := r[n - i]; for i:= 0 step 1 until n-2 do Y[i+2] :=y[n-i]; Y[1]:=Y[2];
                                   U := read; ka:= 1;
                                   Ph1[n] := 1/m × Pr1[n] + U × E/R[n];
                                   GRAMMEL(m, gamma, g, w, n, R, Y, Pr1, Ph1, ds1, dt1);
                                   UITVOER(ka, Pr1, ds1, Ph1, dt1);
                                   GRAFIEK(n, Pr1, Ph1, R, Y)
                             end
                       end;


if comb = 3 then
begin real ka, U, U1, U2; array Pr2, Ph2, ds2, dt2[1 : n];
      NLCR; PRINTTEXT(⊀ combinatie3, voorgeschreven U1 en Prn ⊁); NLCR;
      Ph1[n]:= read;
      GRAMMEL(m, gamma, g, w, n, r, y, Pr1, Ph1, ds1, dt1);
      U1:= r[1] × 1/E × (Ph1[1] - 1/m × Pr1[1]);
      w:= 0; Pr2[n]:= 0; Ph2[n]:= read; ds2[n]:= dt2[n]:= 0;
      GRAMMEL(m, gamma, g, w, n, r, y, Pr2, Ph2, ds2, dt2);
      U2:= r[1] × 1/E × (Ph2[1] - 1/m × Pr2[1]); U:= read;
      ka:= (U - U1)/U2;NLCR; PRINTTEXT(⊀ ka= ⊁); FLOT(5, 3, ka);
      UITVOER2(ka, Pr1, Ph1, Pr2, Ph2, ds1, ds2, dt1, dt2)
;    GRAFIEK(n, Pr1, Ph1, r,y)
end;
```

```
if comb = 4 then
begin real ka, U, U1, U2; array R, Y, Pr2, Ph2, ds2, dt2[1 : n]; integer i;
      NLCR; PRINTTEXT(‹ combinatie4, voorgeschreven Un en Pr1 ›); NLCR;
      U:= read; for i:= 0 step 1 until n - 1 do  R[i + 1]:= r[n - i]; for i:= 0 step 1 until n-2 do Y[i+2]:= y[n-i]; Y[1]:= Y[2];
      Ph1[n]:= read;
      GRAMMEL(m, gamma, g, w, n, R, Y, Pr1,Ph1, ds1, dt1);
      U1:= R[1] × 1/E × (Ph1[1] - 1/m × Pr1[1]);
      w:= 0; Pr2[n]:= 0; Ph2[n]:= read; ds2[n]:= dt2[n]:= 0;
      GRAMMEL(m, gamma, g, w, n, R, Y, Pr2, Ph2, ds2, dt2);
      U2:= R[1] × 1/E × (Ph2[1] - 1/m × Pr2[1]);
      ka:= (U - U1)/U2; NLCR; PRINTTEXT(‹ ka = ›); FLOT(5, 3, ka);
      UITVOER2(ka, Pr1, Ph1, Pr2, Ph2, ds1, ds2, dt1, dt2)
;     GRAFIEK(n, Pr1, Ph1, R, Y)
end;

if comb = 5 then
begin real ka, U, U1, U2, Un; array Pr2, Ph2, ds2, dt2[1 : n];
      NLCR; PRINTTEXT(‹ combinatie5, U1 en Un voorgeschreven ›); NLCR;
      U:= read; Un:= read;
      Ph1[n]:= Un × E/r[n] + 1/m × Pr1[n];
      GRAMMEL(m, gamma, g, w, n, r, y, Pr1, Ph1, ds1, dt1);
      U1:= r[1] × 1/E × (Ph1[1] - 1/m × Pr1[1]);
      Un:= 0; w:= 0; Pr2[n]:= Pr1[n]; ds2[n]:= dt2[n]:= 0;
      Ph2[n]:= 1/m × Pr2[n];
      GRAMMEL(m, gamma, g, w, n, r, y, Pr2, Ph2, ds2, dt2);
      U2:= r[1] × 1/E × (Ph2[1] - 1/m × Pr2[1]);
      ka:= (U - U1)/U2; PRINTTEXT(‹ ka = ›); FLOT(5, 3, ka);
      UITVOER2(ka, Pr1, Ph1, Pr2, Ph2, ds1, ds2, dt1, dt2)
;     GRAFIEK(n, Pr1, Ph1, r,y)
end;


if comb = 6 then
begin real ka, Pr0; array Pr2, Ph2, ds2, dt2[1 : n];
      NLCR; PRINTTEXT(‹ combinatie6, voorgeschreven Pr1 en Prn ›); NLCR;
      Ph1[n]:= read;
      GRAMMEL(m, gamma, g, w, n, r, y, Pr1, Ph1, ds1, dt1);
      w:= 0; Pr2[n]:= 0; Ph2[n]:= read; ds2[n]:= dt2[n]:= 0;
      GRAMMEL(m, gamma, g,w, n, r, y, Pr2, Ph2, ds2, dt2);
      Pr0:= read;
      ka:= (Pr0 - Pr1[1])/Pr2[1]; PRINTTEXT(‹ ka:= ›); FLOT(5, 3, ka);
      UITVOER2(ka, Pr1, Ph1, Pr2, Ph2, ds1, ds2, dt1, dt2)
;GRAFIEK(n,Pr1,Ph1,r,y)
end;
```

```
        if comb = 7 ∨ comb = 8 then
        begin array R, Y[1 : n]; integer i; real ka;
              for i:= 0 step 1 until n - 1 do R[i + 1]:= r[n - i]; for i:= 0 step 1 until n-2 do Y[i+2]:= y[n-i]; Y[1]:= Y[2];
             if comb = 7 then
              begin real U, Us; NLCR; Ph1[n]:= Pr1[n];
                    PRINTTEXT(⊰ combinatie 7, schyf zonder gat, Un voorgeschreven ⊱); NLCR;
                    GRAMMEL(m, gamma, g, w, n, R, Y, Pr1, Ph1, ds1, dt1);
                    for i:=1 step 1 until n do begin Pr1[i]:= Pr1[i] + 1/2 × ds1[i];
                                                      Ph1[i]:= Ph1[i] + 1/2 × dt1[i] end;
                    U:= read; ka:=(U×E/R[1] - Ph1[1] + (1/m)×Pr1[1])/(1-1/m);
                    NLCR; PRINTTEXT(⊰  n        sigma-r          sigma-h ⊱);
                    for i:=1 step 1 until n do  begin NLCR; FIXT(2,0,i); Pr1[i]:= Pr1[i] + ka; SPACE(5); FLOT(5,3,Pr1[i]);
                                                                         Ph1[i]:= Ph1[i] + ka; SPACE(5); FLOT(5,3,Ph1[i]);
end;
              GRAFIEK(n, Pr1, Ph1, R, Y)
             end;

              if comb = 8  then
              begin real Pro; Ph1[n]:= Pr1[n];
                    PRINTTEXT(⊰ combinatie 8, schyf zonder gat, Pr n voorgeschreven⊱); NLCR;
                    GRAMMEL(m, gamma, g, w, n, R, Y, Pr1, Ph1, ds1, dt1);
                    for i:= 1 step 1 until n do begin Pr1[i]:= Pr1[i] + 1/2 × ds1[i];
                                                     Ph1[i]:= Ph1[i] + 1/2 × dt1[i] end;
                    Pro:= read; ka:= Pro - Pr1[1]; NLCR; PRINTTEXT(⊰  n        sigma-r         sigma-h ⊱);
                    for i:=1 step 1 until n do begin NLCR; FIXT(2,0,i); Pr1[i] := Pr1[i] + ka; SPACE(5); FLOT(5,3,Pr1[i] );
                                                               Ph1[i] := Ph1[i] + ka; SPACE(5); FLOT(5,3,Ph1[i]) end;

              GRAFIEK(n, Pr1, Ph1, R, Y)
        end
     end;

    end


  end
 end
 ;end
 progend
```

18