Een Algol-programma voor de exacte oplossing van een periodiek belaste rechthoekige koker (A314)

Een Algol-programma vóor de exacte oplossing van een periodiek belaste rechthoekige koker (A314)

1. Inleiding

Voor een dunwandige rechthoekige koker kan de exacte spanningsverdeling bepaald ten gevolge van een periodieke belasting op het cilindrisch oppervlak door de koker op te vatten als een samenstel van platen. Bovendien dient verondersteld te worden dat de belasting aangrijpt in de hoekpunten of in de middens van de platen. In het binnenkort verschijnende proefschrift van de schrijver van dit rapport "Enige praktische aspecten en theoretische achtergronden van de torsietheorie van Vlasov voor dunwandige rechthoekige kokers" is afgeleid dat de oplossing voor een harmonische belasting gebaseerd is op een stelsel lineaire inhomogene vergelijkingen in acht constanten. Voor de verschillende belastingssystemen, aangeduid met systeem I, II..., VII, zijn de matrixelementen en de rechterleden van dit vergelijkingsstelsel in hoofdstuk 5 van het hiervoor genoemde proefschrift vermeld. Geconstateerd kan worden dat zinvol een indeling te maken is in de systemen I, III, V, ..., VII enerzijds en II en IV anderzijds.

Wanneer de periodieke belasting in een Fourierreeks ontwikkeld wordt kan numeriek slechts een eindig aantal termen (zeg: N) van deze ontwikkeling geaccepteerd worden. Voor iedere component kunnen de acht constanten bepaald worden als oplossing van het vergelijkingenstelsel. Wanneer alle constanten bepaald zijn kunnen de spanningen en vervormingen op een willekeurige plaats bekend worden als resultaat van een eidige sommatie (sommeren tot N).

De in 4 gegeven Algol-tekst is gebaseerd op een periodieke blokbelasting (periode 2a) die symmetrisch is ten opzichte van x = 0 en x = a en antimetrisch ten opzichte van x = $\frac{1}{2}$a (x: axiale coördinaat). Bovendien is het programma zo ingericht dat op eenvoudige wijze de invloed van de periode van een harmonische belasting kan worden geanalyseerd. Bij een harmonische belasting is het uiteraard voldoende de amplitudi van de interessante kracht- en vervormingsgrootheden te berekenen.

In hoofdstuk 6 van het reeds eerder genoemde proefschrift zijn dimensieloze spanningsgrootheden ingevoerd, die uitermate geschikt zijn om als vergelijkingsobjecten te dienen bij het onderzoek naar de waarde van de gemodificeerde Vlasov-theorie. Programma A314 biedt de mogelijkheid deze dimensieloze grootheden te berekenen.

2. In te lezen grootheden

real E	Elasticiteitsmodulus
real nu	Dwarscontractiecoëfficiënt
real bb1, bb2, t1, t2	Afmetingen van een dwarsdoorsnede (b_1, b_2, t_1, t_2)
real F	Oppervlakte van verstijver-dwarsdoorsnede
integer N	Aantal termen van de Fourierreeksontwikkeling dat in rekening wordt gebracht; de keuze N = 1 biedt de mogeli-kheid een harmonische belasting te analyseren
integer M	Aantal gebieden waarin het traject $0 \leq x \leq \frac{1}{2}a$ verdeeld wordt; ieder traject zal worden verdeeld in gelijke intervallen (zie X (1:M).
integer theorie	Theorie = 1 : dimensieloze grootheden worden berekend theorie ≠ 1 : werkelijke spannings- en vervormingsgrootheden worden berekend
integer array X(1:M)	De waarden van X(1), X(2), ..., X(M) bepalen het aantal intervallen waarin ieder van de M gebieden wordt verdeeld
integer Y	Het aantal intervallen waarin $0 \leq y \leq b_1$ (resp. b_2) wordt verdeeld; y is de coördinaat in een dwarsdoorsnede van de koker

integer BEL	BEL kan de waarde 0, 1, ..., 7 gegeven worden; BEL = 1, 2, ..., 7 correspondeert met de keuze van één van de basissystemen I, ..., VII; de amplitudo van de harmonische belasting, respectievelijk de grootte van de blokbelasting wordt Q(i) genoemd; er geldt

BEL = 1, 2, 3 of 4 $\quad Q(i) = r_i \dfrac{b_1}{b_1^2+b_2^2} \quad (i=BEL)$

$Q(i) = 0$ als $i \neq BEL$

BEL = 5 of 6 $\quad Q(i) = \frac{1}{2} r_i \quad (i=BEL)$

$Q(i) = 0 \quad (i \neq BEL)$

BEL = 7 $\quad Q(7) = r_7$

$Q(i) = 0 \quad (i=1,...,6)$

BEL = 0 $\quad$ Q(i) voor i=1, ..., 7 moet worden ingelezen

De keuze BEL = 0 biedt de mogelijkheid de invloed van een systeem dat ontstaat als superpositie van de systemen I, ..., VI te onderzoeken (in dit geval moet Q(7) steeds nul worden gekozen, tenzij Q(1) = ... = Q(6) = 0)

real QQ	Alleen inlezen als BEL ≠ 0; QQ = Q(BEL)
array Q(1:7)	Q(1), ..., Q(7) inlezen (zie BEL)
real ca	Halve blokbreedte van periodieke blokvormige belasting; wanneer N = 1 (harmonische belasting) is de waarde van ca van geen belang; kies dan ca = 0
real a	Halve periode van periodieke of harmonische belasting

3. Toelichting

Het oplossen van het stelsel vergelijkingen gebeurt volgens de methode van Crout. De opgenomen tekst is een (nog) niet officiële versie afkomstig uit de "Rekengroep" van de afdeling Wiskunde THE. Het betreft hier: <u>real procedure</u> inprod (i, i1, i2, ai, bi, c); <u>real procedure</u> CROUTDE - COMPOSITION (n, A, LU, p, interchange, eps, singular); <u>procedure</u> CROUTSOLUTION (n, LU, p, b, x).

Bij de keuze BEL = 0 bestaat, zoals wij in 2 reeds opmerkten, de mogelijkheid een willekeurige combinatie van de belastingssystemen I, ..., VI te analyseren. De gegeven Algol-tekst toont dat hierbij voorzichtig te werk gegaan moet worden, met name bij een belasting waarbij een van de systemen (I, III, V of VI) en (II of IV) betrokken zijn.

In alle gevallen worden de spanningen en vervormingen slechts berekend voor punten gelegen in plaat 1 en plaat 2 waarbij $y_1 < 0$, respectievelijk $y_2 \lessgtr 0$. Bij een belasting volgens een der basissystemen si dit vanwege de heersende symmetrie of antimetrie in het geheel geen beperking. Bij de keuze BEL = 0 waarbij in het algemeen geen symmetrie meer aanwezig is brengt deze beperking informatieverlies te weeg. De tekst kan gewijzigd worden zodat voor ieder punt in een dwarsdoorsnede de spanningen en vervormingen bepaald kunnen worden. Het zal daartoe noodzakelijk zijn stelselvergelijkingen voor ieder van de basissystemen afzonderlijk op te lossen.

In alle gevallen worden de ingelezen grootheden uitgevoerd. Wanneer een harmonische belasting wordt onderzocht zijn alleen de amplitudi interessant. De grootheid x correspondeert voor een periodieke belasting met de waarde van de axiale coördinaat wanneer theorie $\neq 1$. Worden de dimensieloze grootheden bepaald dan is de waarde van x gelijk aan $\alpha_0 x$ (zie proefschrift). Bij een harmonische belasting is $x = 0$ wanneer theorie $\neq 1$, anders krijgt x de waarde $\frac{\alpha_0}{\pi}$.

De grootheid y correspondeert met y_1, respectievelijk y_2 al naar gelang plaat 1 of plaat 2 wordt beschouwd. De grootheden Sx, Sy, Txy, u, v, w, Mx, My, Mxy, Vx en Vy zijn de aanduidingen voor σ_x, σ_y, τ_{xy}, u, v, w, M_x, M_y, M_{xy}, V_x en V_y uit ons proefschrift. Wanneer theorie = 1 corresponderen deze grootheden, met uitzondering van de verplaatsingscomponenten u, v en w, met de dimensieloze spanningsgrootheden uit het reeds meerdere malen genoemde proefschrift (aangeduid met).

Wij merken nog op dat bij een harmonische belasting bovendien de oplossing van het stelsel vergelijkingen wordt gegeven, aangeduid met A1, B1, F1, G1, A2, B2, F2, G2. Deze grootheden zijn gelijk aan de constanten A_1, B_1, E_1, F_1, A_2, B_2, E_2 en F_2 bij de analyse van de belastingssystemen I, III, V, ..., VII en aan C_1, D_1, G_1, H_1, C_2, D_2, G_2 en H_2 wanneer basissystemen II of IV wordt onderzocht.

Nadat een bepaalde koker met een bepaalde belasting numeriek geanalyseerd is, wordt opnieuw een waarde voor a gevraagd. Wanneer a = 0 moet aan alle invoer-grootheden opnieuw een waarde worden toegekend; de keuze a < 0 brengt de beëindiging van de berekeningen met zich mee. Wanneer a < 0 wordt de gegeven waarde als nieuwe halve periode geïnterpreteerd terwijl alle andere grootheden hun waarde behouden.
De hele berekening wordt nogmaals uitgevoerd maar nu met een andere waarde voor a. Op deze wijze is het programma uitermate geschikt om de invloed van de periode op het spannings- en vervormingsveld te onderzoeken.

4. Algol-tekst

4. Algoltekst A 314-6

```
begin procedure PUNCH(x); print (x);
      procedure  FIXP(n,m,x); FIXT(n,m,x);
      procedure  ABSFIXP(n,m,x); ABSFIXT (n,m,x);
      procedure  PUNLCR; NLCR;
      procedure  FLOP (n,m,x); FLOT (n,m,x);
      procedure  PUSPACE (n); SPACE (n);
      procedure  PUTEXT (s); PRINTTEXT (s);
      procedure  STOPCODE; ;
      procedure  RUNOUT; ;
begin comment Periodieke lijnkrachten op rechthoekige koker (ir. Janssen, 28-9-1966)
    Inlezen: E, nu, b1, b2, t1, t2, F(verstijveropp.), N(aantal termen in Fourrierreeks voor blokbelasting, antimetrisch
             t.o.v. x = 0.5 × a), M(aantal gebieden in x-richting), X[1:M] (aantal intervallen per gebied), Y(aantal intervallen
             in y-richting), BEL(type belasting: 0,1,2,3,4,5,6,7), Q[1:7] als BEL = 0 anders QQ, ca(halve blokbreedte),
             a(halve periode).
             Voor amplitudo harmonische lijnbelasting en grootte blokbel. wordt gekozen:
             BEL = 1,2,3 en 4         q[i] × b1/sqrt(b1 ↑ 2 + b2 ↑ 2) = Q[i] = QQ        Q[i] = 0 als i ≠ BEL
             BEL = 5 of 6             q[i] × 0.5 = Q[i] = QQ
             BEL = 7                  q[7] = Q[7] = QQ (steeds los van de rest behandelen)
             BEL = 0                  Q[i] inlezen;
     integer N, M, Y, BEL, n, k, her, i, M1, m, teller, herstel,j, theorie;
     real E, nu, bb1, bb2, t1, t2, F, QQ, G, pi, eps, ca, a, p1, p2, h1, h2, C1, C2, S1, S2, p, D1, D2, h3, x, y, bb, Sx, Sy, Txy,
          u, v, w, Mx, My, Mxy, Vx, Vy, al, ax, ay, ha, hb, hf, hg, haa, hbb, ch, sh, co, si, alf, aa1, ep;
          PUNLCR; PUTEXT (‹Periodieke lijnkrachten op rechthoekige koker volgens exacte theorie›);
  LA: E := read; nu := read; bb1 := read; bb2 := read; t1 := read; t2 := read; F := read; N := read; M := read; theorie := read;
     G := 0.5 × E/(1 + nu); pi := 3.14159265359; eps := ⏨-12; D1 := E × t1 ↑ 3/(12 × (1 - nu ↑ 2)); D2 := E × t2 ↑ 3/(12 × (1 - nu ↑ 2));
     h1 := 4 × E / (bb1 / t1 ↑ 3 + bb2 / t 2 ↑ 3); h1 := h1/(1 - nu ↑ 2); aa1 := 4 × E × (bb1 × bb2) ↑ 2 × (bb1 × t1 + bb2 × t2 + 3 × F )/3;
     alf := sqrt(0.5 × sqrt(h1/aa1)); ep := (bb1 × t2 + bb2 × t1) × sqrt(aa1 × h1)/(32 × G × t1 × t2 × (bb1 × bb2) ↑ 2);
     if theorie = 1 then begin PUNLCR; PUNLCR; PUTEXT(‹Dimensieloze grootheden›); PUNLCR;
     PUTEXT(‹alf, eps›); PUSPACE(5); FLOP(5, 2, alf); FLOP(5, 2, ep) end;
     begin array a1, a2, b1, b2, a3, a4, b3, b4, f1, f2, g1, g2[1:N], Q[1:7], d[1:8, 1:8], r[1:8], a11, b11, f11, g11, f22, g22, a22,
          b22[1:N];
          integer array pivot[1:8], X[1:M];
```

```
boolean interchange;
real procedure sinh(x); value x; real x;
begin real t; t := exp(x); sinh := 0.5 × (t - 1/t) end;
real procedure cosh(x); value x; real x;
begin real t; t := exp(x); cosh := 0.5 × (t + 1/t) end;
real procedure f(ay, P, Q); value ay, P, Q; real ay, P, Q;
begin real t, it; t := exp(ay); it := 1/t; f := 0.5 × (P × (t - it) + ay × Q × (t + it)) end;
real procedure g(ay, P, Q); value ay, P, Q; real ay, P, Q;
begin real t, it; t := exp(ay); it := 1/t; g := 0.5 × (P × (t + it) + ay × Q × (t - it)) end;

real procedure inprod(i, i1, i2, ai, bi,c);  value i1, i2, c;
integer i, i1, i2; real ai, bi,c;
begin real s; s := c;
      for i := i1 step 1 until i2 do  s := s + ai × bi;
      inprod := s
end   inprod;

real procedure CROUTDECOMPOSITION (n, A, LU, p, interchange, eps, singular);
value n, eps;  integer n;  real eps;  array A, LU; integer array p;  boolean interchange;  label singular;
begin integer i,j,k,kk; real q,r,s; array norm[1:n];
      for k:=1 step 1 until n do
      begin s:=0;
            for j:=1 step 1 until n do begin LU[k,j]:=r:=A[k,j]; s:=s+abs(r) end;
            if s=0 then goto singular else norm[k]:=s
      end   computation of rownorms;
      q:=1;
      for k:=1 step 1 until n do
      begin s:=0;
            for i:=k step 1 until n do
            begin LU[i,k]:=r:=inprod  (j, 1, k-1, -LU[i,j], LU[j,k], LU[i,k]);
                                        if i=k ∨ interchange then
                  begin r:=abs(r)/norm[i]; if r > s then begin s:=r; kk:=i  end   end
            end   search for pivot;
            p[k]:=kk;
            if kk ≠ k then
```

```
          begin norm[kk]:= norm[k] ;
                for j:= 1 step 1 until n do
                begin r:= LU[k,j]; LU[k,j]:= LU[kk,j]; LU[kk,j]:= r  end
          end   interchange rows;
          r:= LU[k,k]; if s < eps then goto singular; if s < q then q := s;
          for j:=k+1 step 1 until n do
          begin LU[j,k]:=LU[j,k]/r; LU[k,j]:=inprod(i,1,k-1,-LU[k,i],LU[i,j],LU[k,j]) end
     end k;
     CROUTDECOMPOSITION:= q
end CROUTDECOMPOSITION;

procedure CROUTSOLUTION ( n, LU, p, b, x );
value n; integer n; array LU, b, x; integer array p;
begin integer i, k, kk; real r;
     for k:=1 step 1 until n do x[k]:=b[k];
     for k:=1 step 1 until n do
     begin kk:=p[k]; if kk ≠ k then begin r:=x[k]; x[k]:=x[kk]; x[kk]:=r  end  end ;
     for k:= 2 step 1 until n do
     x[k]:= inprod( i,1,k-1,-LU[k,i],x[i],x[k]);
     for k:=n step -1 until 1 do
     x[k]:= inprod( i,k+1,n,-LU[k,i],x[i],x[k])/LU[k,k]
end CROUTSOLUTION;

interchange := true;
for n := 1 step 1 until M do X[n] := read; Y := read; BEL := read;
if BEL ≠ 0 then QQ := read else for n := 1 step 1 until 7 do Q[n] := read; ca := read;
LB: a := read; if a = 0 then goto LA; if a < 0 then goto END;
PUNLCR; PUNLCR; PUNLCR;
PUTEXT(≮E        nu       b1       b2       t1       t2       F        a        ca       N        BEL   QQ≯);
PUNLCR; FLOP(3, 1, E); PUSPACE(2); FIXP(1, 3, nu); PUSPACE(2); FIXP(3, 3, bb1); PUSPACE(2); FIXP(3, 3, bb2);
PUSPACE(2); FIXP(2, 3, t1); PUSPACE(2); FIXP(2, 3, t2); PUSPACE(2); FIXP(3, 3, F); PUSPACE(2); FIXP(4, 3, a);
PUSPACE(2); FIXP(2, 4, ca);
PUSPACE(2); ABSFIXP(3, 0, N); PUSPACE(2); ABSFIXP(1, 0, BEL); PUSPACE(2);
if BEL ≠ 0 then FLOP(3, 1, QQ)
```

```
          else begin PUNLCR; PUTEXT(≮Q[i] =≯);
                    for n := 1 step 1 until 7 do
                    begin FLOP(3, 1, Q[n]); PUSPACE(2) end
               end;
if BEL ≠ 0 then begin for n := 1 step 1 until 7 do Q[n] := 0; Q[BEL] := QQ end;
teller := 1;
if BEL = 1 ∨ BEL = 3 ∨ BEL > 4 then her := 1;
if BEL = 2 ∨ BEL = 4 then her := 2;
if BEL = 0 then
begin if abs(Q[2]) + abs(Q[4]) < ₁₀-10 then her := 1
                                   else begin if abs(Q[1]) + abs(Q[3]) + abs(Q[5]) + abs(Q[6]) + abs(Q[7]) < ₁₀-10
                                             then her := 2
                                             else begin her := 0; teller := 0; h1 := Q[2]; h2 := Q[4] end
                                        end
end; j := -1;
for n := 1 step 2 until N do
begin p := n × pi/a; p1 := bb1 × p; p2 := bb2 × p;
LC: for m := 1 step 1 until 8 do for k := 1 step 1 until 8 do d[m, k] := 0;
if her = 1 ∨ teller = 0 then begin Q[2] := Q[4] := 0;
                                   C1 := cosh(p1); C2 := cosh(p2);
                                   S1 := sinh(p1); S2 := sinh(p2)
                              end
                    else begin C1 := sinh(p1); C2 := sinh(p2);
                                   S1 := cosh(p1); S2 := cosh(p2)
                              end;
     d[1, 1] := S1; d[1, 2] := 2 × S1/(1 + nu) + p1 × C1; d[1, 5] := -S2; d[1, 6] := -2 × S2/(1 + nu) - p2 × C2;
     d[2, 1] := t1 × C1 + F × p × (1 + nu) × S1; d[2, 5] := t2 × C2 ; d [2, 6] := t2 × (C2 + p2 × S2);
     d[2, 2] := t1 × (C1 + p1 × S1) + F × p × (1 + nu) × (2 × S1/(1 + nu) + p1 × C1);
     d[3, 3] := C1/bb1; d[3, 4] := ( C1 + p1 × S1)/bb1; d[3, 7] := -C2/bb2;
     d[3, 8] := (-C2 - p2 × S2)/bb2; d[4, 3] := D1 × S1/bb1; d[4, 4] := D1 × (2 × S1/(1 - nu) + p1 × C1)/bb1;
     d[4, 7] := D2 × S2/bb2; d [4, 8] := D2 × (2 × S2/(1 - nu) + p2 × C2)/bb2;d [5, 1] := - (1 + nu) × C1;
     d[5, 2] := (1 - nu) × C1 - (1 + nu) × p1 × S1; d[5, 7] := - S2/p2; d[5, 8] := -C2;
     d[6, 3] := - S1/p1 ; d[6, 4] := - C1; d[6, 5] := (1 + nu) × C2;
     d[6, 6] := -(1 - nu) × C2 + (1 + nu) × p2 × S2; d[7, 5] := +t2 × S2; d[7, 6] := + t2 × p2 × C2;
     d[7, 3] := D1 × (1 - nu) × p × C1/(E × bb1); d[7, 4] := (-(1 + nu) × C1 + (1 - nu) × p1 × S1) × D1 × p/(E ×bb1);
```

```
d[8, 1] := -t1 × S1; d[8, 2] := -t1 × p1 × C1; d[8, 7] := C2; d[8, 8] := -(1 + nu) × C2/(1 - nu) + p2 × S2;
h3 := + D2 × (1 - nu) × p/(E × bb2); d[8, 7] := h3 × d[8, 7]; d[8, 8] := h3 × d[8, 8];
a3[n] := (Q[5] - Q[6])/t1 × bb1/bb2; b3[n] := -(1 + nu) × a3[n]/2;
a4[n] := (Q[5] + Q[6])/t2; b4[n] := -(1 + nu) × a4[n]/2;
r[1] := - g(p1, a3[n] + 2 × b3[n]/(1 + nu), b3[n]) + g(p2, a4[n] + 2 × b4[n]/(1 + nu), b4[n]);
r[2] := -t1 × f(p1, a3[n] + b3[n], b3[n]) - t2 × f(p2, a4[n] + b4[n], b4[n]);
r[2] := r[2] - F × p × (1 + nu) × g (p1, a3[n] + 2 × b3[n]/(1 + nu), b3[n]) + Q[7 ]; r[3] := r[4] := 0;
r[5] := (1 + nu) × f (p1, a3 [n] - (1 - nu) × b3 [n] / (1 + nu), b3 [n]);
r[6] := -(1 + nu) × f (p2, a4[n] - (1 - nu) × b4 [n]/(1 + nu), b4[n]);
r[7] := t2 × g(p2, a4[n], b4[n]); r[8] := t1 × g (p1, a3[n], b3[n]);
r[7] := -r[7] - (Q[1] + Q[2] - Q[3] - Q[4]) × bb2/bb1;
r[8] := r[8] + (Q[1] + Q[2] + Q[3] + Q[4]);
CROUTDECOMPOSITION(8, d, d, pivot, interchange, eps, singular);
CROUTSOLUTION(8, d, pivot, r, r);
if N = 1 then h3 := 1 else begin j := (if abs( Q [7]) < ₁₀-10 then 1 else - j); h3 := 4 × j × sin(n × pi × ca/a)/
(pi × n) end;
if teller = 0 then begin a11[n] := h3 × r[1]; b11[n] := h3 × r[2];
                         f11[n] := h3 × r[3]; g11[n] := h3 × r[4];
                         a22[n] := h3 × r[5]; b22[n] := h3 × r[6];
                         f22[n] := h3 × r[7]; g22[n] := h3 × r[8];
                         teller := 1;  Q[2] := h1; Q[4] := h2; Q [1] := Q [3] := Q [5] := Q [6] := Q[ 7]:= 0; goto LC
                    end
               else begin a1[n] := h3 × r[1]; b1[n] := h3 × r[2];
                          f1[n] := h3 × r[3]; g1[n] := h3 × r[4];
                          a2[n] := h3 × r[5]; b2[n] := h3 × r[6];
                          f2[n] := h3 × r[7]; g2[n] := h3 × r[8];
                    end;
a3[n] := h3 × a3[n]; b3[n] := h3 × b3[n]; a4[n] := h3 × a4[n]; b4[n] := h3 × b4[n];
if N = 1 then
begin PUNLCR; PUTEXT(⫷a1           b1          f1          g1          a2          b2          f2          g2⫸);
      PUNLCR; FLOP(4, 2, a1[n]); PUSPACE(1); FLOP(4, 2, b1[n]); PUSPACE(1); FLOP(4, 2, f1[n]); PUSPACE(1);
      FLOP(4, 2, g1[n]); PUSPACE(1); FLOP(4, 2, a2[n]); PUSPACE(1); FLOP(4, 2, b2[n]); PUSPACE(1);
      FLOP(4, 2, f2[n]); PUSPACE(1); FLOP(4, 2, g2[n])
end
```

```
end alle constanten zijn bepaald;
comment berekening spanningen enz.;
for i := 1 step 1 until M do
begin if i = 1 then M1 := 0 else M1 := 1;
      for m := M1 step 1 until X[i] do
      begin if X[i] = 0 then x := 0 else x := m × a/(2 × M × X[i]) + (i - 1) × a/(2 × M);
            teller := 1;
         LD: herstel := 1; for k := 0 step 1 until Y do
            begin if teller = 1 then bb := bb1 else bb := bb2;
                  y := k × bb/Y; Sx := Sy := Txy := u := v := w := Mx := My := Mxy := Vx := Vy := 0;
                  for n := 1 step 2 until N do
                  begin al := n × pi/a; ax := al × x; ay := al × y;
                        if teller = 1 then begin ha := a1[n]; hb := b1[n];
                                                 hf := f1[n]; hg := g1[n];
                                                 haa := a3[n]; hbb := b3[n]
                                           end
                                      else begin ha := a2[n]; hb := b2[n];
                                                 hf := f2[n]; hg := g2[n];
                                                 haa := a4[n]; hbb := b4[n]
                                           end;if her = 0 then haa := hbb := 0;
                     LE: if her = 1 then begin ch := cosh(ay); sh := sinh(ay) end
                                    else begin ch := sinh(ay); sh := cosh(ay) end;
                        si := sin(ax); co := cos(ax);
                        if N = 1 then co := si := 1;
                        Sx := Sx + ((ha + 2 × hb) × sh + ay × hb × ch + (haa + 2 × hbb) × ch + ay × hbb × sh) × co;
                        Sy := Sy - (ha × sh + ay × hb × ch + haa × ch + ay × hbb × sh) × co;
                        Txy := Txy + ((ha + hb) × ch + ay × hb × sh + (haa + hbb) × sh + ay × hbb × ch) × si;
                        u := u + ((ha + 2 × hb/(1 + nu)) × sh + ay × hb × ch + (haa + 2 × hbb/(1 + nu)) × ch +
                        ay × hbb × sh) × si/al;
                        v := v - ((ha - (1 - nu) × hb/(1 + nu)) × ch + ay × hb × sh + (haa - (1 - nu) × hbb/
                        (1 + nu)) × sh + ay × hbb × ch) × co/al;
                        w := w + (hf × sh + ay × hg × ch) × co/(al ↑ 2 × E × bb);
                        Mx := Mx + ((hf - 2 × nu × hg/(1 - nu)) × sh + ay × hg × ch) × co;
                        My := My - ((hf + 2 × hg/(1 - nu)) × sh + hg × ay × ch) × co;
```

```
          Mxy := Mxy + ((hf + hg) × ch + ay × hg × sh) × si;
          Vx := Vx + ((hf + 2 × (2 - nu) × hg/(1 - nu)) × sh + ay × hg × ch) × al × si;
          Vy := Vy + ((hf - (1 + nu) × hg/(1 - nu)) × ch + ay × hg × sh) × al × co;
          if her = 0 then begin if teller = 1 then begin ha := a11[n]; hb := b11[n];
                                                        hf := f11[n]; hg := g11[n];
                                                        haa := a3[n]; hbb := b3[n]
                                                  end
                                             else begin ha := a22[n]; hb := b22[n];
                                                        hf := f22[n]; hg := g22[n];
                                                        haa := a4[n]; hbb := b4[n]
                                                  end;
                               her := 1; herstel := 0; goto LE
                           end;
          if herstel = 0 then her := 0
      end;
      if teller = 1 then h3 := D1/bb1 else h3 := D2/bb2; h3 := h3 × (1 - nu)/E;
      Mx := h3 × Mx; My := h3 × My; Mxy := h3 × Mxy; u := u × (1 + nu) /E; v := v × (1 + nu)/E; w := w/E;
      Vx := h3 × Vx; Vy := h3 × Vy;
      if theorie = 1 ∧ BEL ≠ 0 then
      begin   if teller = 1 then h1 := t1 else h1 := t2;
              if BEL < 5 then h2 := 8 × bb2 × QQ;
              if BEL = 5 ∨ BEL = 6 then h2 := 4 × bb1 × QQ;
              if BEL = 7 then h2 := bb1 × bb2 × QQ × 4 × alf;
              if N ≠ 1 then begin  x := x × alf;  h2 := h2 × 2 × ca end
              else begin x := alf/al; h2 := h2/alf end;
              Sx := Sx × alf × aa1/(E  × bb1 × bb2 × h2); Sy := Sy × alf ×aa1/(E ×bb1 × bb2× h2);
              Txy := Txy × 8 × bb1 × bb2 × h1/h2; h1 := 8/(alf × h2);
              Mx := Mx × h1; My := My × h1; Mxy := Mxy × h1; h1 := h1/bb;
              Vx := Vx × h1; Vy := Vy × h1;
      end;
      if k = 0 then
      begin PUNLCR; PUNLCR;
            if teller = 1 then begin PUTEXT(⟨x =⟩); FIXP(4, 3, x); PUNLCR;PUNLCR;
PUTEXT(⟨y       Sx        Sy        Txy        u        v        w        Mx        My        Mxy        Vx        Vy⟩);
```

```
                                   PUNLCR; PUTEXT(⊰           Plaat 1⊱)
                                end
                             else begin PUNLCR; PUNLCR; PUTEXT(⊰          Plaat 2⊱); PUNLCR end
                  end;
                  PUNLCR; FIXP(2, 2, y); FLOP(4, 2, Sx); FLOP(4, 2, Sy); FLOP(4, 2, Txy); FLOP(4, 2, u); FLOP(4, 2, v);
                  FLOP(4, 2, w); FLOP(4, 2, Mx); FLOP(4, 2, My); FLOP(4, 2, Mxy); FLOP(4, 2, Vx); FLOP(4, 2, Vy)
               end;
               teller := teller + 1; if teller = 2 then goto LD
            end
         end;
         goto LB;
    singular: PUNLCR; PUTEXT(⊰Matrix is singulier⊱); goto END
      end;
   END:
   end
   end
```

oktober 1966
ir. J.D. Janssen