

Serie AP 400
=====

Inhoud d.d. juni 1964
=====

AP 400	Programma SYMEVE, TJD 050662
AP 401	Programma Factor-analyse, TJD 010762
AP 402	Programma Eigenwaarden en Principale Componenten, TJD 061162
AP 403	Programma Multipеле Correlaties, Eigenwaarden en Principale Componenten, A. Dirkzwager en T.J. Dekker, 261062
AP 404	Programma Priemgetallenkleedje, TJD 051062
AP 405	Programma Reele Nulpunten van Polynomen d.m.v. ZEREX, TJD 050763
AP 406	Programma REVAVEC, TJD 050863

Serie AP 400
=====

Inhoud d.d. april 1964
=====

AP 400	Programma SYMEVE, TJD 050662
AP 401	Programma Factor-analyse, TJD 010762
AP 402	Programma Eigenwaarden en Principale Componenten, TJD 061162
AP 403	Programma Multipelle Correlaties, Eigenwaarden en Principale Componenten, A.Dirkwager en T.J.Dekker, 261062

Serie AP 400

Deze serie bevat volledige ALGOL-60 programma's, bruikbaar in het MC-ALGOL-systeem.

De tekst van elk programma wordt voorafgegaan door een commentaar, bevattende:

- 1) de namen der gebruikte MCP's, d.w.z. die MCP's die in het programma met name worden genoemd;
- 2) een beknopte beschrijving van het rekenproces en eventueel methode, mogelijkheden of beperkingen;
- 3) de benodigde gegevens, die via de band, het consolewoord enz. moeten worden meegegeven;
- 4) een beschrijving van de door het programma geleverde output;
- 5) eventueel de toelaatbare omvang van de invoergegevens en een schatting van de benodigde tijd.

In de tekst van elk programma worden ter afkorting de bodies van de gebruikte procedures uit de serie AP 200 vervangen door:

begin < body AP ... > end,

met dien verstande, dat in plaats van ... het nummer van de desbetreffende procedure wordt vermeld.

comment

AP 400

Programma SYMEVE, TJD 050662. Gebruikt de MCP's SUM, EVEN en FIXT.

Het programma berekent de eigenwaarden en eigenvectoren van symmetrische matrices. De tolerantie voor de eigenwaarden-iteratie is 10^{-11} \times de in het programma berekende matrix-norm. Voor verdere details zie de commentaren der gebruikte procedures.

De getallenband moet de matrices leveren, elke matrix gegeven door orde $n \geq 1$ en daarna de rijen van de onderdriehoek, dwz van elke rij de elementen t/m het hoofddiagonaal-element. De band moet worden afgesloten met 0, gevolgd door een brokscheider.

Het programma typt de eigenwaarden in 12 decimalen drijvende-komma notatie en de eigenvectoren in 11 decimalen achter de vaste komma. De eigenvectoren zijn zo genormeerd, dat de Euclidische norm = 1 is. Van de eigenvectoren komen ten hoogste 10 elementen op een regel. Bovendien zorgt het programma voor pagina-indeling. Elke pagina bevat 66 regels, waarvan er ten hoogste 56 gebruikt worden. Men zorg er voor, dat aan het begin van het programma het papier in de schrijfmachine ongeveer vijf regels voorbij de perforatie gedraaid is.

Naast het programma moet in het geheugen ruimte beschikbaar zijn voor ongeveer $0.5 \times n \times (n + 9)$ getallen van type real en een integer n-vector, waarbij n de orde voorstelt. Voor een X1 rekenmachine met een geheugencapaciteit van 12288 woorden betekent dit, dat $n = 80$ nog toegestaan is.

De tijd, benodigd voor een matrix van orde 30, is ongeveer 30 minuten;

begin integer i, j, n, lico;

```
procedure privec (i, k, l, a); value k, l; integer i, k, l; real a;
begin integer m; m:= 2 + (l - k) : 10;
      for i:= k step 1 until l do
        begin if i = k then begin NLCR; k:= k + 10 end; FIXT (1, 11, a) end;
        NLCR; lico:= lico + m;
        if lico + m > 56 then nl (66 - lico)
      end privec;
```

```
procedure nl (i); value i; integer i;
begin for i:= i - 1 while i  $\geq$  0 do begin NLCR; lico:= lico + 1 end;
      if lico  $\geq$  66 then lico:= 0
end nl;
```

```

real procedure ZERO (x, a, b, fx, e);
value a, b; real x, a, b, fx; array e;
begin < body AP 200 > end ZERO;

procedure PSP1 (A, i, j, n, B, BB, D);
value n; integer i, j, n; real A; array B, BB, D;
begin < body AP 210 > end PSP1;

procedure PREP (D, B, n, eps, E);
value n; integer n; real eps; array D, B, E;
begin < body AP 211 > end PREP;

real procedure SEIVA (D, BB, n, n1, n2, k, E);
value n; integer n, n1, n2, k; array D, BB, E;
begin < body AP 212 > end SEIVA;

procedure SEIVC (D, B, n, n1, n2, x, V);
value n, n1, n2, x; integer n, n1, n2; real x; array D, B, V;
begin < body AP 213 > end SEIVC;

procedure TRASF1 (A, i, j, n, B, V);
value n; integer i, j, n; real A; array B, V;
begin < body AP 214 > end TRASF1;

procedure SYMEVE (A, i, j, n, eps, OVA, OVEC);
value n; integer i, j, n; real A, eps; procedure OVA, OVEC;
begin < body AP 215 > end SYMEVE;

START: lico:= 0; goto OUT;
IN: begin integer array P[1 : n]; array C[1 : n × (n + 1) : 2];
      j:= n × (n + 1) : 2; for i:= 1 step 1 until j do C[i]:= read;
      if lico > 0 ∧ lico + n × (2 + (n - 1) : 10) > 53 then nl (66 - lico) else nl (3);
      for j:= 1 step 1 until n do P[j]:= (j - 1) × j : 2;
      SYMEVE (C[P[j] + i], i, j, n, 10-11, print, privec)
    end;
OUT: n:= read; if n ≠ 0 then goto IN
end

```

comment AP 401

Programma Factor-analyse, TJD 010762. Gebruikt de MCP's EVEN, SUM, INPROD en de ponsende MCP's RUNOUT, PUNLCR, FIXP, PUTTEXT, TAPEND en STOPCODE.

Op de getallenband moeten gegeven zijn: het aantal variabelen, het aantal individuen en daarna van elk individu de vector der ruwe scores, zijnde gehele getallen.

Het programma pons: het aantal individuen, van elke variabele de som der scores en de som der score-quadraten, daarna de orde en de onderdriehoek van de correlatie-matrix en tenslotte de eigenwaarden ≥ 0.5 met de bijbehorende principale componenten (zo genormeerd, dat $\text{quadraatsom} = \text{eigenwaarde}$). Het programma typt: de orde van de correlatie-matrix (= het aantal variabelen) en daarna, telkens op een nieuwe regel, van elke berekende eigenwaarde de volgende gegevens: het rangnummer, de eigenwaarde en, indien gewijzigd, de nieuwe waarde van $E[2]$. Zodra de volgende eigenwaarde kleiner is dan de drempel 0.5, worden de eigenwaarden met rangnummer $\leq E[2]$ overgeslagen.

Steeds geldt, dat de komende eigenwaarden met rangnummer $\leq E[2]$ monotoon afnemen. Als $E[2]$ gelijk is aan de orde, zijn dus alle komende eigenwaarden monotoon afnemend.

Voor verdere details, zie de commentaren der gebruikte procedures.

Het programma is geschreven voor een X1 met een levend geheugen van 12.288 woorden. Het aantal variabelen mag niet groter zijn dan 72. Voor een X1 met een kleiner levend geheugen moet de constante 6000 (voorkomende aan het begin van het programma, na de label start) worden vervangen door een kleiner getal of door available min 500. Voor het aantal individuen geldt geen beperking. De score-matrix wordt, zo nodig, bij gedeelten ingelezen.

De benodigde tijd voor een systeem met 45 variabelen en 120 individuen was 30 minuten voor de berekening van de correlatie-matrix en 90 minuten voor de berekening van 17 eigenwaarden en principale componenten. Voor een systeem met 24 variabelen en 162 individuen was de totale tijd 30 a 45 minuten;

begin integer i, j, k, m, n, rest, length, N; real labda; array e[1 : 2], E[0 : 7];

real procedure ZERO (x, a, b, fx, e);
value a, b; real x, a, b, fx; array e;
begin < body AP 200 > end ZERO;


```

procedure SPAP (A, i, j, n, B, BB, D, E);
value n; integer i, j, n; real A; array B, BB, D, E;
begin < body AP 231 > end SPAP;

```

```

real procedure SEIGENVA (D, BB, n, e, E);
value n; integer n; array D, BB, e, E;
begin < body AP 232 > end SEIGENVA;

```

```

procedure SEIGENVEC (D, B, n, E, V);
value n; integer n; array D, B, E, V;
begin < body AP 233 > end SEIGENVEC;

```

```

procedure STRASF (A, i, j, n, B, V);
value n; integer i, j, n; real A; array B, V;
begin < body AP 234 > end STRASF;

```

```

procedure SEVAVEC (A, i, j, n, e, E, OVA, OVEC);
value n; integer i, j, n; real A; array e, E; procedure OVA, OVEC;
begin < body AP 235 > end SEVAVEC;

```

```

procedure OVA (x);
real x;
begin integer k, n2;
      NLCCR; PUNLCCR;
      labda:= x; k:= E[0]; n2:= E[2]; if E[4] < 0.5 then E[0]:= n2;
      print (k); print (labda); if E[0] = E[1] then print (n2); FIXP (2, 5, labda)
end OVA;

```

```

procedure OVEC (V);
array V;
begin integer k;
      labda:= sqrt (labda); PUNLCCR;
      for k:= 1 step 1 until m do FIXP (1, 4, V[k] × labda); PUNLCCR; TAPEND
end OVEC;

```

```

start:  m:= read; length:= (m + 1) × m : 2; N:= read; n:= (6000 - 2 × length) : m - 2;
begin  array C[1 : length]; integer array P, S[1 : m];
      for j:= 1 step 1 until m do begin P[j]:= (j - 1) × j : 2; S[j]:= 0 end;
      for j:= 1 step 1 until length do C[j]:= 0; rest:= N;
in:    if rest < n then n:= rest;
      begin integer array X[1 : n, 1 : m];
            for k:= 1 step 1 until n do
goin:  for j:= 1 step 1 until m do X[k, j]:= read;
            for j:= 1 step 1 until m do
            begin S[j]:= S[j] + SUM (k, 1, n, X[k, j]);
                  for i:= 1 step 1 until j do
                    C[P[j] + i]:= C[P[j] + i] + INPROD (k, 1, n, X[k, i], X[k, j])
                  end
            end; rest:= rest - n; if rest ≠ 0 then goto in; RUNOUT;
      PUNLCR; PUTEXT (⟨Aantal individuen ⟩); FIXP (8, 0, N);
      PUNLCR; PUTEXT (⟨Som der scores⟩); PUNLCR;
      for j:= 1 step 1 until m do FIXP (8, 0, S[j]);
      RUNOUT; PUNLCR; PUTEXT (⟨Som der score-quadraten⟩); PUNLCR;
      begin array V[1 : m];
            for j:= 1 step 1 until m do
            begin V[j]:= sqrt (C[P[j] + j] × N - S[j] 2); FIXP (13, 0, C[P[j] + j) end;
            PUNLCR; TAPEND;
cormat: PUNLCR; PUTEXT (⟨Correlatie-matrix van de orde ⟩); FIXP (3, 0, m); PUNLCR;
      for j:= 1 step 1 until m do
      begin PUNLCR; for i:= 1 step 1 until j do
            begin C[P[j] + i]:= (C[P[j] + i] × N - S[i] × S[j]) / (V[i] × V[j]);
                  FIXP (1, 4, C[P[j] + i))
            end; PUNLCR
      end; TAPEND
      end;
princ comp: NLCR; print (m); NLCR;
      PUNLCR; PUTEXT (⟨Eigenwaarden en principale componenten⟩); PUNLCR;
      e[1]:= 0; e[2]:= 10-6; SEVAVEC (C[P[j] + i], i, j, m, e, E, OVA, OVEC); STOPCODE
end
end

```


comment AP 402

Programma Eigenwaarden en Principale Componenten, TJD 061162. Gebruikt de MCP's EVEN, SUM en de ponsende MCP's RUNOUT, PUNLCR, FIXP, PUTTEXT, TAPEND en STOPCODE.

Op de getallenband moeten zijn gegeven: de orde en de onderdriehoek van de correlatie-matrix rij voor rij. XEEN $(-0)/2 \wedge 26$ is de drempel voor de te berekenen eigenwaarden (deze voldoet dus aan: $-1 < \text{drempel} < 1$).

Het programma pons: de orde van de correlatie-matrix en daarna de eigenwaarden \geq drempel met de bijbehorende principale componenten (zo genormeerd, dat $\text{quadraatsom} = \text{eigenwaarde}$). Het programma typt: de orde van de correlatie-matrix en daarna, telkens op een nieuwe regel, van elke berekende eigenwaarde de volgende gegevens: het rangnummer, de eigenwaarde en, indien gewijzigd, de nieuwe waarde van E[2] en tenslotte, telkens als de drempel bereikt is, de drempel. Zodra de volgende eigenwaarde kleiner is dan de drempel, worden de eigenwaarden met rangnummer \leq E[2] overgeslagen.

Steeds geldt, dat de komende eigenwaarden met rangnummer \leq E[2] monotoon afnemen. Als E[2] gelijk is aan de orde, zijn dus alle komende eigenwaarden monotoon afnemend.

Voor verdere details, zie de commentaren der gebruikte procedures.

Voor een X1 met 12.288 woorden moet de orde van de correlatie-matrix kleiner dan 80 zijn. De benodigde tijd voor de berekening van 12 eigenwaarden en principale componenten van een correlatie-matrix van de orde 32 bedroeg 40 minuten;

```
begin   integer i, j, m, length; real labda, drempel; array e[1 : 2], E[0 : 7];
```

```
   real procedure ZERO (x, a, b, fx, e);  
   value a, b; real x, a, b, fx; array e;  
   begin   < body AP 200 >   end ZERO;
```

```
   procedure SPAP (A, i, j, n, B, BB, D, E);  
   value n; integer i, j, n; real A; array B, BB, D, E;  
   begin   < body AP 231 >   end SPAP;
```

```

real procedure SEIGENVA (D, BB, n, e, E);
value n; integer n; array D, BB, e, E;
begin < body AP 232 > end SEIGENVA;

procedure SEIGENVEC (D, B, n, E, V);
value n; integer n; array D, B, E, V;
begin < body AP 233 > end SEIGENVEC;

procedure STRASF (A, i, j, n, B, V);
value n; integer i, j, n; real A; array B, V;
begin < body AP 234 > end STRASF;

procedure SEVAVEC (A, i, j, n, e, E, OVA, OVEC);
value n; integer i, j, n; real A; array e, E; procedure OVA, OVEC;
begin < body AP 235 > end SEVAVEC;

procedure OVA (x);
real x;
begin integer k, n2;
      NLCCR; PUNLCCR;
      labda:= x; k:= E[0]; n2:= E[2]; print (k); print (labda);
      if E[0] = E[1] then print (n2);
xeen:  drempel:= XEEN (-0)/67108864;
      if E[4] < drempel then begin E[0]:= n2; print (drempel) end;
      FIXP (2, 5, labda)
end OVA;

procedure OVEC (V);
array V;
begin integer k;
      labda:= sqrt (labda); PUNLCCR;
      for k:= 1 step 1 until m do FIXP (1, 4, V[k] × labda); PUNLCCR; TAPEND
end OVEC;

```

```

start:  m:= read; length:= (m + 1) × m : 2;

       begin   array C[1 : length]; integer array P[1 : m];
              for j:= 1 step 1 until m do P[j]:= (j - 1) × j : 2;
              for j:= 1 step 1 until length do C[j]:= read;

       princ comp: NLCR; print (m); NLCR;

              RUNOUT; PUTTEXT (⟨Orde =⟩); FIXP (3, 0, m); PUNLCR; RUNOUT;
              PUNLCR; PUTTEXT (⟨Eigenwaarden en principale componenten⟩); PUNLCR;

              e[1]:= 0; e[2]:= 10-6; SEVAVEC (C[P[j] + i], i, j, m, e, E, OVA, OVEC); STOPCODE

       end
end

```

programma verandert.

in de gegeven tekst de drempel van de
correlatie- en/of multiple correlatie-coëfficiënt.

als schakel van de
(commutatieve)

comment AP 403

de kwadraten van de

Programma Multipele Correlaties en Eigenwaarden en Principale Componenten. A. Dirkzwager en T. J. Dekker, d.d. 261062. Gebruikt de MCP's EVEN, SUM en de ponsende MCP's RUNOUT, PUNLCR, FIXP, PUTEXT, TAPEND en STOPCODE.

Op de getallenband moeten zijn gegeven: de orde en de onderdriehoek van de correlatie-matrix rij voor rij. XEEN (-0)/2 \wedge 26 is de drempel van de te berekenen eigenwaarden (deze voldoet dus aan: $-1 < \text{drempel} < 1$).

Het programma pons: de squared multiple correlations en daarna de eigenwaarden \geq drempel met de bijbehorende principale componenten (zo genormeerd, dat quadraatsom = eigenwaarde). Het programma typt: de orde van de correlatie-matrix en daarna, telkens op een nieuwe regel, van elke berekende eigenwaarde de volgende gegevens: het rangnummer, de eigenwaarde en, indien gewijzigd, de nieuwe waarde van E[2] en tenslotte, telkens als de drempel bereikt is, de drempel. Zodra de volgende eigenwaarde kleiner is dan de drempel, worden de eigenwaarden met rangnummer \leq E[2] overgeslagen.

Steeds geldt, dat de komende eigenwaarden met rangnummer \leq E[2] monotoon afnemen. Als E[2] gelijk is aan de orde, zijn dus alle komende eigenwaarden monotoon afnemend.

Voor verdere details, zie de commentaren der gebruikte procedures.

Voor een X1 met 12.288 woorden moet de orde van de correlatie-matrix kleiner dan of gelijk aan 55 zijn. De benodigde tijd voor een correlatie-matrix van de orde 45 was 20 minuten voor de berekening van de multipele correlaties en 90 minuten voor de berekening van 17 eigenwaarden en principale componenten;

begin integer i, j, m; real labda, drempel; array e[1 : 2], E[0 : 7];

real procedure SYMDET (A, i, j, n);
value n; integer i, j, n; real A;
begin < body AP 220 > end SYMDET;

procedure syminv (A, i, j, n);
value n; integer i, j, n; real A;
begin < body AP 223 > end syminv;

real procedure ZERO (x, a, b, fx, e);
value a, b; real x, a, b, fx; array e;
begin < body AP 200 > end ZERO;

```

procedure SPAP (A, i, j, n, B, BB, D, E);
value n; integer i, j, n; real A; array B, BB, D, E;
begin < body AP 231 > end SPAP;

real procedure SEIGENVA (D, BB, n, e, E);
value n; integer n; array D, BB, e, E;
begin < body AP 232 > end SEIGENVA;

procedure SEIGENVEC (D, B, n, E, V);
value n; integer n; array D, B, E, V;
begin < body AP 233 > end SEIGENVEC;

procedure STRASF (A, i, j, n, B, V);
value n; integer i, j, n; real A; array B, V;
begin < body AP 234 > end STRASF;

procedure SEVAVEC (A, i, j, n, e, E, OVA, OVEC);
value n; integer i, j, n; real A; array e, E; procedure OVA, OVEC;
begin < body AP 235 > end SEVAVEC;

procedure OVA (x);
real x;
begin integer k, n2;
      NLCR; PUNLCR;
      labda:= x; k:= E[0]; n2:= E[2]; print (k); print (labda);
      if E[0] = E[1] then print (n2);
xeen: drempel:= XEEN (-0)/67108864;
      if E[4] < drempel then begin E[0]:= n2; print (drempel) end;
      FIXP (2, 5, labda)
end OVA;

procedure OVEC (V);
array V;
begin integer k;
      labda:= sqrt (labda); PUNLCR;
      for k:= 1 step 1 until m do FIXP (1, 4, V[k] × labda); PUNLCR; TAPEND
end OVEC;

```



```

start:  m:= read;

begin  array C[1 : m, 1 : m];
        for j:= 1 step 1 until m do for i:= 1 step 1 until j do C[i, j]:= C[j, i]:= read;

mult corr: SYMDET (C[j, i], i, j, m); syminv (C[j, i], i, j, m);
           RUNOUT; PUTTEXT (⟨Squared multiple correlations⟩); PUNLCR;
           for j:= 1 step 1 until m do begin C[j, j]:= 1 - 1/C[j, j]; FIXP (1, 4, C[j, j]) end;
           PUNLCR; TAPEND;

princ comp: NLCR; print (m); NLCR;

           PUNLCR; PUTTEXT (⟨Eigenwaarden en principale componenten⟩); PUNLCR;
           e[1]:= 0; e[2]:= 10-6; SEVAVEC (C[i, j], i, j, m, e, E, OVA, OVEC); STOPCODE

end

end

```


Addendum bij het comment van programma AP 403

Het programma vervangt vóór de eigenwaardenbepaling de diagonaalelementen van de correlatiematrix door de kwadraten van de multipele correlatiecoëfficiënten (als schatting van de communaliteiten).

Programma Priemgetallenkleedje, TJD 051062. Gebruikt de MCP's FACTOR, REMAINDER en de ponsende MCP's RUNOUT, PUNLCR, PUTEXT, PUSPACE, FIXP, ABSFIXP en STOPCODE.

In de vorm van een tableau wordt een afbeelding gegeven van de punten van het complexe vlak, die corresponderen met een deelverzameling van de priemgetallen van Gauss. De afbeeldingen, die worden beschouwd, zijn ingebed binnen een rechthoek en vormen een zgn. priemgetallenkleedje (in analogie met het ontwerp van Prof. dr. Balthasar van der Pol).

Op de te lezen getallenband moeten vier gehele getallen worden gegeven, nl. reeel deel a en imaginair deel b van de startwaarde (aangevend de linker bovenhoek van het kleedje) en breedte c en lengte d van het kleedje. De getallen a, b, c en d moeten, wegens de beperkte integer-capaciteit, zo gekozen worden, dat alle complexe getallen z van het gewenste deel van de ring van Gauss gelegen zijn binnen de cirkel: $\text{modulus}(z) < 2\sqrt{13}$. Verder kiese men breedte en lengte positief en de breedte niet groter dan 74.

Het programma pons de startwaarde en daarna het kleedje van de priemgetallen van het gewenste deel van de ring van Gauss, m.a.w. de priemgetallen z van de ring van Gauss die voldoen aan: $a < \text{Re}(z) < a + c$ en $b < \text{Im}(z) < b + d$. De niet-priemgetallen worden voorgesteld door twee spaties. Van de priemgetallen worden het laatste cijfer van reeel en imaginair deel gepons, waarbij steeds een oneven cijfer voorop staat. Het programma eindigt met de procedure stop. Men kan desgewenst doorstarten met BVA en verkrijgt dan het aansluitende kleedje (met startwaarde $a + (b + d) \times i$). Voor een kleedje startend in de oorsprong (dus $a = b = 0$), met breedte 50 en lengte 200, was de benodigde tijd 12 minuten;

```
begin integer a, b, c, d, f, n, p, q, s, x; real t;
start: RUNOUT; PUTEXT (Picture of Gaussian prime numbers starting from); PUSPACE (3);
a:= read; b:= read; FIXP (4, 0, a); FIXP (4, 0, b); ABSFIXP (1, 0, 15);
c:= read + a - 1; d:= read + b - 1; PUNLCR; PUNLCR; RUNOUT;
entry: for q:= b step 1 until d do
  begin PUNLCR; s:= 0; t:= 0; x:= REMAINDER (abs (q), 10); if REMAINDER (x, 2)  $\neq$  0 then x:= 10  $\times$  x;
  row: for p:= a step 1 until c do
    begin n:= p  $\times$  p + q  $\times$  q; f:= FACTOR (n); if REMAINDER (f, 4) = 3 then f:= f  $\times$  f; if f  $\neq$  n
    then
      notpr: begin if s < 0 then begin ABSFIXP (-s, 0, t); t:= 0; s:= -2 end; s:= s + 2 end
    else
      prime: begin if s > 0 then begin PUSPACE (s); s:= 0 end;
              t:= 100  $\times$  t + (REMAINDER (abs (p), 10)  $\times$  (if x < 10 then 10 else 1) + x);
              s:= s - 2; if p = c then ABSFIXP (-s, 0, t)
            end
    end
  end
end; STOPCODE; stop; q:= d + 1; d:= q + d - b; b:= q; goto entry
end
```

comment AP 405

Programma ter berekening van reële nulpunten van polynomen d.m.v. ZEREX, TJD 050763. Gebruikt de MCP PRINTTEXT en de ponsende MCP's RUNOUT, PUNLCR, FLOP en FIXP.

Op de getallenband moeten zijn gegeven: de relatieve en absolute tolerantie voor de nulpunten en daarna de polynomen, ieder gegeven door graad > 0 gevolgd door de coëfficiënten, de coëfficiënt van x \uparrow graad voorop. (Deze eerste coëfficiënt mag niet 0 zijn.) De band moet worden afgesloten met 0.

Het programma typt: de gegeven relatieve en absolute tolerantie en vervolgens van elk polynoom de graad, de coëfficiënten, de berekende nulpunten, de bijbehorende polynoom-waarden en het benodigde aantal iteraties.

Als XEEN (1) = 1, worden de iteraties, de bijbehorende waarden van het gedeelde polynoom en de iteratie-telling geponst.

Als het aantal iteraties voor een nulpunt de waarde 100 bereikt, wordt de berekening opgegeven en het volgende polynoom aangevat.

De nulpunten van de gegeven polynomen moeten alle reëel en goed van elkaar gescheiden zijn. Voor verdere bijzonderheden zie het commentaar van ZEREX (= AP 236);

```

begin   integer k, n, t; real x, xa; array e[1 : 2];

real procedure ZEREX (x, fx, xa, e); real x, fx, xa; array e;
begin   < body AP 236 >   end ZEREX;

real procedure ZERO (x, a, b, fx, e); value a, b; real x, a, b, fx; array e;
begin   < body AP 230 >   end ZERO;

real procedure POL (A, k, n, x); value n, x; integer k, n; real A, x;
begin   < body AP 237 >   end POL;

real procedure PROD (k, a, b, fk); value a, b; integer k, a, b; real fk;
begin   < body AP 202 >   end PROD;

real procedure MAX (k, a, b, fk); value a, b; integer k, a, b; real fk;
begin   < body AP 201 >   end MAX;

start:  e[1]:= read; e[2]:= read; NLCR; print (e[1]); print (e[2]); goto out;
in:     begin   array A, W[0 : n];

        real procedure F (x, EX); value x; real x; label EX;
        begin   integer i; real r;
                if XEEN (i) = 1 then
                begin   if t = 0 then begin if k = 1 then RUNOUT; PUNLCR end;
                        PUNLCR; FLOP (12, 3, x)
                end; t:= t + 1; if t = 100 then goto EX;
                F:= r:= POL (A[i], i, n, x) / PROD (i, 1, k - 1, x - W[i]);
                if XEEN (i) = 1 then begin FLOP (12, 3, r); FIXP (3, 0, t) end
        end F;

        for k:= 0 step 1 until n do A[k]:= read; xa:= MAX (k, 1, n, abs (A[k] / A[0])  $\uparrow$  (1/k)); x:= 4 × xa + 3;
        NLCR; NLCR; PRINTTEXT (⟨Graad =   ⟩); print (n); NLCR; NLCR;
        PRINTTEXT (⟨Coefficients      Nulpunten      Functie-waarden      Aantal iteraties⟩);
        NLCR; print (A[0]); for k:= 1 step 1 until n do
        begin   integer i;
                NLCR; print (A[k]); t:= 0; print (ZEREX (x, F (x, out), xa, e));
                W[k]:= x; print (POL (A[i], i, n, x)); print (t)
        end   end;
out:     n:= read; if n > 0 then goto in
end

```


comment AP 406

Programma REVAVEC, TJD 050863. Gebruikt de MCP's SUM, INPROD en de ponsende MCP's RUNOUT, PUNLCR, FLOP, FIXP, ABSFIXP, TAPEND en STOPCODE.

Het programma berekent de reële eigenwaarden en eigenvectoren van asymmetrische matrices. Alle eigenwaarden moeten reëel en goed van elkaar gescheiden zijn. De afgeleverde eigenwaarden zijn monotoon afnemend. Voor verdere bijzonderheden zie de commentaren van de gebruikte procedures.

Op de getallenband moeten gegeven zijn: de relatieve tolerantie voor de transformatie (relatief t.o.v. de matrixnorm, een geschikte waarde is 10^{-12}) en de relatieve en absolute tolerantie voor de eigenwaarden, daarna de matrices rij voor rij, elke matrix voorafgegaan door de orde. De band moet worden afgesloten met 0.

Het programma pons de toleranties (in de gegeven volgorde) en daarna van elke matrix: de orde en vervolgens, telkens op een regel, de eigenwaarde, het aantal iteraties voor eigenwaarde en eigenvector en de bereikte precisie voor de eigenvector en daaronder de eigenvector zelf (zo genormeerd, dat de Euclidische norm = 1 is). Onder "bereikte precisie" voor de eigenvector wordt hier verstaan: de Euclidische norm van (getransformeerde matrix $H - \text{eigenwaarde} \times I$) \times eigenvector van H , zijnde de waarde van $E[5]$ (zie REIGENVEC (= AP 240)).

Naast het programma moet in het geheugen ruimte beschikbaar zijn voor ongeveer $1.5 \times n \times (n + 3)$ getallen van type real en een Boolean n -vector, waarbij n de orde is. Voor een X1 met 12 288 woorden betekent dit, dat de orde ≤ 45 moet zijn.

Voor een matrix van de orde 12 was de benodigde tijd circa 15 minuten;

```

begin   integer i, j, n; array e[0 : 2], E[0 : 5];

real procedure ZEREX (x, fx, xa, e); real x, fx, xa; array e;
begin       < body AP 236 >       end ZEREX;

real procedure ZERO (x, a, b, fx, e); value a, b; real x, a, b, fx; array e;
begin       < body AP 230 >       end ZERO;

real procedure PROD (k, a, b, fk); value a, b; integer k, a, b; real fk;
begin       < body AP 202 >       end PROD;

procedure APAP (A, n, eps, norm, B); value n, eps; integer n; real eps, norm; array A, B;
begin       < body AP 238 >       end APAP;

real procedure REIGENVA (A, n, B, e, E, Z, V); value n; integer n; array A, B, e, E, Z, V;
begin       < body AP 239 >       end REIGENVA;

procedure REIGENVEC (A, n, B, e, E, V); value n; integer n; array A, B, e, E, V;
begin       < body AP 240 >       end REIGENVEC;

procedure ATRASF (A, n, B, V); value n; integer n; array A, B, V;
begin       < body AP 241 >       end ATRASF;

procedure REVAVEC (A, n, e, E, Z, OVA, OVEC); value n; integer n; array A, e, E, Z; procedure OVA, OVEC;
begin       < body AP 242 >       end REVAVEC;

procedure OVA (x); real x;
begin   PUNLCR; PUNLCR; FLOP (12, 3, x); ABSFIXP (3, 0, E[3]) end OVA;

procedure OVEC (V); array V;
begin   integer k;
        ABSFIXP (3, 0, E[4]); FLOP (12, 3, E[5]); PUNLCR;
        for k:= 1 step 1 until n do FIXP (1, 11, V[k])
end OVEC;

start:  RUNOUT; for i:= 0, 1, 2 do begin e[i]:= read; FLOP (12, 3, e[i]) end; goto out;
in:     begin   array A[1 : n, 1 : n], Z[1 : n];
        for i:= 1 step 1 until n do for j:= 1 step 1 until n do A[i, j]:= read;
        PUNLCR; PUNLCR; ABSFIXP (3, 0, n); REVAVEC (A, n, e, E, Z, OVA, OVEC)
end;
out:    PUNLCR; TAPEND; n:= read; if n > 0 then goto in; STOPCODE
end

```