stichting
mathematisch
centrum

MC

REKENAFDELING NR 23/72 JANUARI

P.A. BEENTJES
EEN ALGOL 60 VERSIE VAN GESTABILISEERDE
RUNGE-KUTTA METHODEN

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat 49, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Voorwoord

Dit rapport bevat een ALGOL 60 programma voor het oplossen van begin-waarde-problemen.
Het programma en de voorbeelden zijn getest op de Electrologica X8 computer van het Mathematisch Centrum te Amsterdam.
De schrijver wil zijn dank betuigen aan Dr. P.J. van der Houwen voor diens waardevolle suggesties.

Inhoud

1. Inleiding

Er zijn verschillende eenstapsmethoden ontwikkeld voor de numerieke integratie van beginwaarde-problemen van het type

$$(1.1) \qquad \frac{dU}{dt} = H(U,t),\ U = U_0,\ t = t_0,$$

waarin H een (vector) functie is van U en de variabele t en U een (vector) functie is van t.
Als de functie H(U,t) gemakkelijk een aantal malen gedifferentieerd kan worden, kan men voor het oplossen van (1.1) polynoommethoden gebruiken. Zie hiervoor bijvoorbeeld [2] en [3].
In gevallen waarin differentiatie van het rechterlid van (1.1) moeilijk is, worden dikwijls Runge-Kutta methoden gebruikt.
Dit rapport geeft een ALGOL 60 versie van expliciete Runge-Kutta formules met een orde van nauwkeurigheid p, p = 1, 2, 3, die beschreven staan in [1].

In de volgende sectie geven we de genererende matrices voor deze formules. Verder geven we een aantal schattingen voor afbreekfouten.
De derde sectie geeft een beschrijving van het ALGOL 60 programma, de procedure modified runge kutta.
In sectie vier bespreken we een aantal testresultaten.
Allereerst behandelen we de vergelijking van Rayleigh waarbij we als nevenresultaat de oplossing van Van der Pol's vergelijking vinden op een meer efficiënte manier dan in [6].
Met het tweede testvoorbeeld, een hyperbolische differentiaalvergelijking, geven we aan hoe onze procedure gebruikt kan worden bij het oplossen van partiële differentiaalvergelijkingen.
Het derde testvoorbeeld, een drie-lichamen probleem in twee dimensies, kozen we om de noodzaak en het voordeel van een goede stapkeuze-strategie aan te tonen.

2. De theorie van gestabiliseerde Runge-Kutta schema's

In deze sectie behandelen we de theoretische achtergronden van de procedure modified runge kutta. Voor een volledige theoretische beschouwing van gestabiliseerde Runge-Kutta methoden verwijzen we naar [1].

2.1. Algemene structuur van het Runge-Kutta schema

We geven de algemene n-punts formule

$$
(2.1)\qquad \begin{cases} u_0 = U_0 , \\ u_{k+1} = u_k + \theta_0 r_k^{(0)} + \ldots + \theta_{n-1} r_k^{(n-1)}, \quad k = 0, 1, 2, \ldots, \\ r_k^{(0)} = \tau_k \, H(t_k, u_k), \\ r_k^{(j)} = \tau_k \, H(t_k + \mu_j \tau_k, \; u_k + \lambda_{j0} r_k^{(0)} + \ldots + \lambda_{jj-1} r_k^{(j-1)}), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j = 1, 2, \ldots, n-1, \end{cases}
$$

waarin u_k de numerieke oplossing is op het tijdstip t_k.
Verder geldt per definitie $\tau = \tau_k = t_{k+1} - t_k$.
Schema (2.1) kan worden gekarakteriseerd door de matrix

$$
(2.2)\qquad R = \begin{pmatrix} \mu_1 & \lambda_{10} & 0 & \cdots & 0 \\ \mu_2 & \lambda_{20} & \lambda_{21} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \mu_{n-1} & \lambda_{n-1\,0} & \lambda_{n-1\,1} & \cdots & \lambda_{n-1\,n-2} \\ \theta_0 & \theta_1 & \theta_2 & \cdots & \theta_{n-1} \end{pmatrix}.
$$

2.2. Stabiliteit

Beschouw het volgende lineaire beginwaarde-probleem

$$\frac{dU}{dt} = DU,\ U = U_0,\ t = t_0, \tag{2.3}$$

waarin D een matrix is met eigenwaarden λ_i waarvoor geldt

$$Re(\lambda_i) \leq 0,\ \forall i.$$

We definiëren het stabiliteitspolynoom $P_n(z)$ als volgt

$$P_n(z) = 1 + \beta_1 z + \ldots + \beta_n z^n,$$

waarin de parameters β_j, $j = 1, 2, \ldots, n$, functies zijn van de Runge-Kutta parameters μ_1 en λ_{1i} uit (2.2).
Met behulp van het stabiliteitspolynoom kunnen we een stabiel schema opstellen voor de berekening van de oplossing van (2.3)

$$\begin{cases} u_0 = U_0\ , \\ u_{k+1} = P_n(\tau_k D)u_k,\ \tau_k \leq \dfrac{\beta(n)}{\sigma(D)}, & k = 0, 1, 2, \ldots\ . \end{cases}$$

Hierin is $\beta(n)$ een functie van de parameters β_j, $j = 1, 2, \ldots, n$.
$\sigma(D)$ is de spectraalradius van de matrix D.
Zoals bekend kunnen we deze lineaire stabiliteitstheorie toepassen op de locaal-gelineariseerde voorstelling van (1.1).

2.3. Genererende matrix van Runge-Kutta schema's van de orde p, p = 1, 2, 3

De coëfficiënten van de genererende matrix (2.2) kunnen voor de gevallen p = 1, 2, 3 zo gekozen worden dat bij berekeningen volgens het bijbehorende Runge-Kutta schema een beperkt gebruik van geheugenruimte gemaakt wordt.

We geven nu onze versie van deze coëfficiënten uitgedrukt in de parameters β_j, $j = 1, \ldots, n$

$$
(2.4)\qquad
\begin{cases}
\theta_0 = \begin{cases} 0, & p = 1, 2, \\ \frac{1}{4}, & p = 3, \end{cases} \\
\theta_j = 0, \qquad 1 \le j \le n-2, \\
\theta_{n-1} = 1 - \theta_0, \\
\mu_{n-j} = \dfrac{\beta_{j+1}}{\theta_{n-1}(-\theta_0)^{j-1} + \sum_{p=0}^{j-2}(-\theta_0)^p \beta_{j-p}}, \qquad j = 2, \ldots, n-1, \\
\mu_{n-1} = \dfrac{\beta_2}{\theta_{n-1}}, \\
\lambda_{jl} = \begin{cases} \theta_0, & 2 \le j \le n-1,\ l = 0, \\ \mu_1, & j = 1,\ l = 0, \\ \mu_j - \theta_0, & 2 \le j \le n-1,\ l = j-1, \\ 0, & 3 \le j \le n-1,\ 1 \le l \le j-2. \end{cases}
\end{cases}
$$

Voor de gevallen $p = 1, 2$ krijgen we op deze manier

$$\mu_j = \lambda_{jj-1} = \frac{\beta_{n+1-j}}{\beta_{n-j}}, \qquad j = 1, 2, \ldots, n-1,$$

met in het bijzonder $\mu_{n-1} = \frac{1}{2}$ als $p = 2$.

Voor $p = 3$ krijgen we $\mu_{n-1} = \frac{2}{3}$, $\mu_{n-2} = \frac{8}{15}$, $\lambda_{n-1\ n-2} = \frac{5}{12}$ en $\lambda_{n-2\ n-3} = \frac{17}{60}$.

De Runge-Kutta schema's bepaald door (2.4) vereisen slechts de geheugenruimte van drie arrays terwijl bij een zwak gekoppeld stelsel van differentiaalvergelijkingen de benodigde geheugenruimte zelfs tot twee arrays beperkt kan worden.

2.4. Formules voor een schatting van de locale fout

In deze subsectie geven we een aantal formules voor de gewichten θ'_j, $j = 0, 1, \ldots, n'-1$, $n' \leq n$, die gebruikt kunnen worden voor de volgende schatting van de afbreekfout

$$\rho_k = \theta'_0 r_k^{(0)} + \ldots + \theta'_{n'-1} r_k^{(n'-1)} .$$

De parameter λ_{jj-1} duiden we in het vervolg aan met λ_j, $j = 2, \ldots, n-1$.

We geven allereerst de gewichten θ'_j, $j = 0, 1$, voor p^e orde-n-punts-Runge-Kutta schema's ($p=1,2$; $n=2,3$)

$$\theta'_0 = -\theta'_1 = -\frac{b_1}{\mu_1} . \tag{2.5}$$

Voor eerste orde exacte schema's geldt $b_1 = \frac{1}{2} - \beta_2$, voor tweede orde exacte schema's $b_1 = \frac{1}{2}$.

De gewichten θ'_j, $j = 0, 1, 2, 3$, voor p^e orde-n-punts-Runge-Kutta schema's ($p=1,2,3$; $4 \leq n \leq 7$) zijn

$$\text{(2.6)} \quad \begin{cases} \theta'_3 = \dfrac{\mu_2(\mu_1-\mu_2)b_1 - \mu_1\lambda_2[\mu_1(\frac{1}{2}-\beta_2)-b_2]}{\mu_2^2\lambda_3(\mu_1-\mu_2)-\mu_1\mu_3\lambda_2(\mu_1-\mu_3)} , \\ \theta'_2 = \dfrac{b_1-\mu_2\lambda_3\theta'_3}{\mu_1\lambda_2} , \\ \theta'_1 = \dfrac{\frac{1}{2}-\beta_2-\mu_2\theta'_2-\mu_3\theta'_3}{\mu_1} , \\ \theta'_0 = -\theta'_1 - \theta'_2 - \theta'_3 . \end{cases}$$

Voor b_1 en b_2 geldt

$$b_1 = \begin{cases} \frac{1}{6} - \theta_{n-1}\lambda_{n-1}\mu_{n-2}, & p = 1, 2, \\ \frac{1}{6}, & p = 3, \end{cases}$$

$$b_2 = \begin{cases} \frac{1}{3} - \theta_{n-1}\mu_{n-1}^2, & p = 1, 2, \\ \frac{1}{3}, & p = 3. \end{cases}$$

Tot slot geven we de gewichten θ_j', $j = 0, \ldots, 7$, voor p^e orde-n-punts-Runge-Kutta schema's ($p=1,2,3$; $n \geq 8$).

In vectornotatie wordt θ' gegeven door

$$\theta' = (W)^{-1}b, \tag{2.7}$$

waarin W de volgende 8×8-matrix is

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_6 & \mu_7 \\ 0 & 0 & \mu_1\lambda_2 & \mu_2\lambda_3 & \mu_3\lambda_4 & \mu_4\lambda_5 & \mu_5\lambda_6 & \mu_6\lambda_7 \\ 0 & \mu_1^2 & \mu_2^2 & \mu_3^2 & \mu_4^2 & \mu_5^2 & \mu_6^2 & \mu_7^2 \\ 0 & 0 & 0 & \mu_1\lambda_2\lambda_3 & \mu_2\lambda_3\lambda_4 & \mu_3\lambda_4\lambda_5 & \mu_4\lambda_5\lambda_6 & \mu_5\lambda_6\lambda_7 \\ 0 & 0 & \mu_1^2\lambda_2 & \mu_2^2\lambda_3 & \mu_3^2\lambda_4 & \mu_4^2\lambda_5 & \mu_5^2\lambda_6 & \mu_6^2\lambda_7 \\ 0 & 0 & \mu_1\mu_2\lambda_2 & \mu_2\mu_3\lambda_3 & \mu_3\mu_4\lambda_4 & \mu_4\mu_5\lambda_5 & \mu_5\mu_6\lambda_6 & \mu_6\mu_7\lambda_7 \\ 0 & \mu_1^3 & \mu_2^3 & \mu_3^3 & \mu_4^3 & \mu_5^3 & \mu_6^3 & \mu_7^3 \end{pmatrix}.$$

Voor de vector b geldt

$$b = \begin{pmatrix} 0 \\ \frac{1}{2} - \beta_2 \\ \frac{1}{6} - \beta_3 \\ \frac{1}{3} - \theta_{n-1}\mu_{n-1}^2 \\ \frac{1}{24} - \beta_4 \\ \frac{1}{12} - \theta_{n-1}\lambda_{n-1}\mu_{n-2}^2 \\ \frac{1}{8} - \theta_{n-1}\mu_{n-1}\lambda_{n-1}\mu_{n-2} \\ \frac{1}{4} - \theta_{n-1}\mu_{n-1}^3 \end{pmatrix}.$$

3. De procedure modified runge kutta

In de volgende subsectie geven we de heading van modified runge kutta en de betekenis van de parameters.

3.1. Heading en parameters van modified runge kutta

```
procedure modified runge kutta (t, te, m0, m, u, sigma, i,
                                derivative, k, data, alfa, norm,
                                aeta, reta, eta, rho, output);

integer    m0, m, i, k, norm;
real       t, te, sigma, alfa, aeta, reta, eta, rho;
array      u, data;
procedure  derivative, output;
```

Parameters:

t : <variable>;
t wordt gebruikt als Jensen parameter;
bij een aanroep van modified runge kutta moet t de begin-waarde van de onafhankelijke variabele hebben;

te : <expression>;
de eindwaarde van t;

m0, m : <expression>;
indices van de eerste en laatste vergelijking van het stelsel differentiaalvergelijkingen;

u : een een-dimensionaal array u[m0:m];
bij een aanroep van modified runge kutta moet dit array de startwaarden van $U(t_0)$ bevatten;

sigma : <expression>;
grootste absolute waarde van de eigenwaarden van de Jacobiaan D van het stelsel die niet in het positieve halfvlak liggen;
sigma moet door de gebruiker worden gegeven;

i : <variable>;
telt het aantal evaluaties van H(U,t) tijdens een integratiestap en loopt daarbij van 0 tot en met n-1;
i kan als Jensen parameter gebruikt worden bij hyperbolische differentiaalvergelijkingen;

derivative: een procedure die als volgt door de gebruiker moet worden gegeven:
procedure derivative (x,v); real x; array v;
<body>;
na de uitvoering van deze procedure moet het array v de componenten van H(U,t) bevatten;

k : <variable>;
telt het aantal integratiestappen;
bij de eerste aanroep van modified runge kutta dient k 0 te zijn; er wordt dan gestart met een stap

$$\tau_0 = \min\left[\frac{\eta_0}{||H(U_0,t_0)||}, \frac{\beta(n)}{\sigma(D_0)}\right];$$

bij volgende aanroepen van de procedure begint het integratieproces, als k niet weer op 0 gezet wordt, met een stapgrootte die gebaseerd is op drie laatst berekende discrepanties uit de vorige aanroep;

data : een een-dimensionaal array data[-3: data[-2]];
data[-3]: het aantal evaluaties, n', van de afgeleide H(U,t) dat gebruikt wordt voor een schatting van de locale fout die gemaakt wordt in een integratiestap;

data[-2]: het aantal evaluaties, n, van de afgeleide H(U,t) dat gebruikt wordt om de vectoren $r_k^{(j)}$ te berekenen;

data[-1]: orde van nauwkeurigheid, p, van de methode;

data [0]: stabiliteitsparameter $\beta(n)$;

data [1], ..., data[data[-2]]: coëfficiënten β_j van het polynoom dat de locale stabiliteit garandeert;

alfa : <expression>;
met deze parameter kan de gebruiker de toename van de stap-
grootte als volgt regelen:

$$\tau_k \leq \text{alfa} * \tau_{k-1} \ ;$$

norm : <expression>;
voor norm = 1 wordt door modified runge kutta gerekend met
de maximumnorm, voor norm = 2 met de Euclidische norm;

aeta, reta: <expression>;
verlangde absolute en relatieve locale precisie;
als aeta en reta beide negatief zijn wordt door modified
runge kutta niet aan nauwkeurigheidscondities gedaan hetgeen
inhoudt dat er geen locale fouten berekend worden en de
staplengte als volgt bepaald wordt

$$\tau_k = \beta(n) \ / \ \sigma(D_k);$$

eta : <variable>;
de tolerantie η_k die een functie is van aeta, reta en $||U_k||$;

rho : <variable>;
de discrepantie die gebruikt wordt voor een schatting van de
locale fout die gemaakt is in de laatst genomen integratie-
stap;

output : een procedure die door de gebruiker moet worden gegeven:
procedure output;
<body>;
in deze procedure kan men tot uitvoer gelasten van bijvoor-
beeld
t, u[m0], ..., u[m], sigma, k, eta, rho;

In de volgende subsecties geven we de body van modified runge kutta en een bespreking van de procedures die daarin voorkomen.

3.2. De body van modified runge kutta

```
begin i:=-1;
    begin integer p,n,n1,q;
        own real ec0,ec1,ec2,tau0,tau1,tau2,taus,t2;
        real theta0,thetanm1,tau,gamma,betan;
        array mu,labda[0:data[-2]-1],beta[1:data[-2]],
                thetha[0:data[-3]-1],ro,r[m0:m];
        boolean start,step1;

        real procedure normfunction(nrm,w); integer nrm;
        array w;
        begin integer j; real s,x;
            s:=0;
            if nrm=1 then for j:=m0 step 1 until m do
            begin x:=abs(w[j]);
                if x>s then s:=x
            end;
            if nrm=2 then s:=sqrt(vecvec(m0,m,0,w,w));
            normfunction:=s
        end normfu;

        procedure coefficient;
        begin integer j,k;real s;
            n1:=data[-3];n:=data[-2];p:=data[-1];
            betan:=data[0];gamma:=.5;
            for j:=1 step 1 until n do beta[j]:=data[j];
            thetanm1:=if p=3 then .75 else 1;
            theta0:=1-thetanm1;
            mu[n-1]:= beta[2]/thetanm1;
            labda[n-1]:=mu[n-1]-theta0;
            labda[0]:=mu[0]:=0;
            s:=thetanm1;
            if n>2 then
            for j:=n-2 step -1 until 1 do
            begin s:=beta[n-j]-s×theta0;
                mu[j]:=beta[n+1-j]/s;
                labda[j]:=mu[j]-theta0
            end;
            if n1=2 then
            begin thetha[0]:=if p=1 then (beta[2]-.5)/labda[1]
                else -.5/labda[1];
                thetha[1]:=-thetha[0];
                q:=2
            end;
            if n1=4 then
            begin real a,b,c,d,e,f;
                e:=if p=3 then 0 else beta[3];
                f:=if p=3 then 0 else beta[2];
                a:=mu[1];b:=mu[2];c:=mu[3];d:=theta0;
```

```
        thetha[3]:=(b×(a-b)×(1/6-e)-a×(b-d)×(a×(.5-beta[2])
          -(1/3-f×f)))/(b×(a-b)×(c-d)×b-a×c×(b-d)×(a-c));
        thetha[2]:=d:=(1/6-e-b×(c-d)×thetha[3])/(a×(b-d));
        thetha[1]:=c:=(.5-beta[2]-b×d-c×thetha[3])/a;
        thetha[0]:=-c-d-thetha[3];
        q:=if p=1 then 2 else 3
end;
if n1=8 then
begin real array alpha[1:6,1:6],aux[0:3];
      for j:=1 step 1 until 6 do
      for k:=1 step 1 until 6 do
      alpha[j,k]:=if j=1 then
        beta[n+1-k]/beta[n-k] else
        if j=2 then mu[k+1]-mu[1] else
        if j=3 then (if k=1 then 0 else
        beta[n+2-k]/beta[n-k]) else
        if j=4 then mu[k]×alpha[1,k] else
        if j=5 then mu[k+1]×alpha[1,k] else
        mu[k+1]×alpha[2,k];
      thetha[1]:=1/6-beta[3];
      thetha[2]:=1/3-beta[2]×mu[n-1]-mu[1]×(.5-beta[2]);
      thetha[3]:=1/24-beta[4];
      thetha[4]:=1/12-beta[3]×mu[n-2];
      thetha[5]:=1/8-beta[3]×mu[n-1];
      thetha[6]:=.25-beta[2]×mu[n-1]↑2
        -mu[1]×(1/3-beta[2]×mu[n-1]);
        aux[0]:=10-12;rnksolelm(alpha,6,aux,thetha);
        s:=(.5-beta[2]-sum(j,1,6,thetha[j]))/mu[1];
        for j:=7 step -1 until 2 do
        thetha[j]:=thetha[j-1]/mu[j];
        thetha[1]:=s;thetha[0]:=-sum(j,1,7,thetha[j]);
        q:=p+1
    end
end coefficient;

procedure local error bound;
eta:=aeta+reta×normfunction(norm,u);

procedure local error construction;
if i < n1-1 then
begin integer j; real tht;
    if i=0 then
    for j:=m0 step 1 until m do ro[j]:=0;
    tht:=thetha[i]×tau;
    for j:=m0 step 1 until m do
    ro[j]:=ro[j]+tht×r[j];
    if i=n1-1 then
    begin i:=n-1;rho:=normfunction(norm,ro);i:=n1-1;
        ec0:=ec1;ec1:=ec2;ec2:=rho/tau↑q
    end
end l.e.c.;
```

```
procedure stepsize;
begin real tauacc,taustab,aa,bb,cc,ec;
    local error bound;
    if eta>0 then
    begin if start then
        begin if k=0 then
            begin integer j;
            for j:=m0 step 1 until m do
            r[j]:=u[j];
            i:=0; derivative(t,r);
            tauacc:=eta/normfunction(norm,r);
            step1:=true
        end else
        if step1 then
        begin tauacc:=(eta/rho)↑(1/q)×tau2;
            if tauacc>10×tau2 then
            tauacc:=10×tau2 else step1:=false
        end else
        begin bb:=(ec2-ec1)/tau1; cc:=ec2-bb×t2;
            ec:=bb×t+cc;
            tauacc:=if ec<0 then tau2 else
            (eta/ec)↑(1/q);
            start:=false
        end
end else
    begin aa:=((ec0-ec1)/tau0+(ec2-ec1)/tau1)
                /(tau1+tau0);
        bb:=(ec2-ec1)/tau1-aa×(2×t2-tau1);
        cc:=ec2-t2×(bb+aa×t2); ec:=cc+t×(bb+t×aa);
        tauacc:=if ec<0 then
        taus else (eta/ec)↑(1/q);
        if tauacc>alfa×taus then tauacc:=alfa×taus;
        if tauacc<gamma×taus then
        tauacc:=gamma×taus;
        if tauacc<10-12 × t then tauacc:= 10-12 × t
    end
end else tauacc:=te-t;
taustab:=betan/sigma;if taustab<10-12×t then
begin printtext(‹stable time-step less than 10-12×t›);
    goto end of modified rk
end;
tau:=if tauacc>taustab then taustab else tauacc;
    taus:=tau; if tau>te-t then tau:=te-t;
    tau0:=tau1;tau1:=tau2;tau2:=tau
end stepsize;

procedure difference scheme;
begin integer j;
    real mt,lt;
next term:
    mt:=mu[i+1]×tau;lt:=labda[i+1]×tau;
```

```
for j:=m0 step 1 until m do r[j]:=u[j]+1t×r[j];
i:=i+1;derivative(t+mt,r);
if eta>0 then local error construction;
if (i=0∧p=3)∨i=n-1 then
begin real tht;
      tht:=if i=0 then theta0×tau else thetanm1×tau;
      for j:=m0 step 1 until m do
      u[j]:=u[j]+tht×r[j]
      end;
      if i<n-1 then goto next term;
      t2:=t;t:=t+tau
   end diff.sch.;

   start:=k=0;
   coefficient;
 next level:
   stepsize;k:=k+1;i:=-1;difference scheme;output;
   if t<te then goto next level
end;
end of modified rk:
end modified rk;
```

3.3. De procedure coefficient

In deze procedure worden allereerst de parameters θ_0, θ_{n-1}, λ_j en μ_j, j = 1, 2, ..., n-1, berekend volgens de formules (2.4). Verder worden de coëfficiënten $\theta^!_j$, j = 0, 1, ..., n'-1, berekend volgens de formules (2.5), (2.6) en (2.7).

Tot slot wordt een parameter q berekend (die afhankelijk is van n'), waarvan de betekenis duidelijk zal worden in sectie 3.7.

3.4. De procedure difference scheme

In deze procedure worden de waarden van u[j], j = m0, ..., m, de componenten van de numerieke oplossing u_k, vervangen door de componenten van u_{k+1}. Tijdens de berekening van u_{k+1} wordt na elke evaluatie van $r_k^{(i)}$, tevens een schatting van de discrepantie (locale fout) opgebouwd door de procedure local error construction.

3.5. De procedure local error construction

De schatting van de locale fout $\rho_k = \sum_{i=0}^{n'-1} \theta^!_i r_k^{(i)}$ wordt in deze procedure in n' stappen berekend. Daarna wordt $||\rho_k||$ berekend in een

norm die door de gebruiker is opgegeven. ρ_k kan de gedaante hebben van een laatste correctieterm (l.c.t.) of van een eerst verwaarloosde term (e.v.t.).
De l.c.t. is de term met τ^p in de Taylor-ontwikkeling van u_{k+1} uit (2.1).
De e.v.t. is het verschil van de termen τ^{p+1} in de Taylor-ontwikkelingen van u_{k+1} uit (2.1) en de locaal analytische oplossing door het punt (t_k, u_k).

3.6. De procedure local error bound

De tolerantie η_k die een maat geeft voor de maximaal toelaatbare waarde van de discrepantie $||\rho_k||$ in de k^e integratiestap, wordt in deze procedure als volgt berekend.

$$\text{eta} = \text{aeta} + \text{reta} * ||u_k||.$$

3.7. De procedure stepsize

In deze procedure proberen we een integratiestap τ_k te bepalen zodanig dat geldt $\eta_k = ||\rho_k||$. Omdat $||\rho_k||$ pas bekend is na de k^e integratiestap, hebben we een strategie nodig om τ_k te voorspellen. Hiertoe zijn we er van uitgegaan dat in de omgeving van (t_k, τ_k) de fout $||\rho||$ zich als volgt als een functie van t en τ gedraagt

$$(3.1) \qquad ||\rho|| = f(t,\tau,A,B,C,\ldots),$$

waarin f een gegeven functie is en A, B, C, ... parameters zijn die bepaald worden door de vergelijkingen

$$(3.2) \qquad f(t_j,\tau_j,A,B,C,\ldots) = ||\rho_j(\tau_j)||, \qquad j = k-1,\ k-2,\ k-3,\ \ldots .$$

De k^e integratiestap τ_k wordt berekend door de vergelijking

$$(3.3) \qquad f(t_k,\tau_k,A,B,C,\ldots) = \eta_k$$

op te lossen.

We hebben de volgende representaties van $||\rho||$ beschouwd

$$||\rho|| = C\tau^q, \tag{3.4}$$

$$||\rho|| = (Bt+C)\tau^q, \tag{3.4'}$$

$$||\rho|| = (At^2+Bt+C)\tau^q. \tag{3.4"}$$

In deze voorstellingen van $||\rho||$ wordt de afhankelijkheid met betrekking tot τ van de uitdrukkingen tussen haakjes, de zogenaamde "foutconstante", verwaarloosd. We kunnen nu met deze voorstellingen, ingevuld in (3.3), τ_k expliciet bepalen als de foutconstante positief is. Als dit laatste niet zo is dan blijkt onze voorstelling van $||\rho||$ te falen. In dat geval kiezen we $\tau_k = \tau_{k-1}$.

In de formules (3.4), (3.4') en (3.4") heeft q de waarde p + 1 als we ρ schatten met de e.v.t. terwijl q = p als we ρ schatten met de l.c.t.

We kunnen echter de eerste stap τ_0 niet berekenen met een formule van het type (3.4) omdat $||\rho_{-1}||$ niet bestaat.

Daarom hebben we voor τ_0 een ruwe schatting gemaakt die overeenkomt met een (pessimistische) stapgrootte van een schema van orde nul,

$$\tau_0 = \frac{\eta_0}{||r_0^{(0)}||}.$$

De volgende stap τ_1 volgt nu uit (3.4),

$$\tau_1 = \tau_0 \sqrt[q]{\frac{\eta_1}{||\rho_0||}}.$$

De stapvoorspelling is in dit stadium echter nog primitief, zodat de kans bestaat dat τ_1 een veel te grote waarde krijgt. Dit kan gebeuren als $||\rho_0||$ zeer klein is ten gevolge van een kleine tijdstap τ_0. We hebben ons op het standpunt gesteld dat een stapgrootte realistisch genoemd kan worden, als de laatst berekende discrepantie groter is dan 10^{-q} * tolerantie (als een volgende stap dus 10 * groter is dan de vorige, verwachten we dat de discrepantie groter zal zijn dan de

toleratie). Daarom hebben we in deze fase van het integratieproces de volgende strategie aangehouden

1) als $\tau_k > 10\tau_{k-1}$ dan wordt τ_k gelijk gemaakt aan $10\tau_{k-1}$; voor de bepaling van τ_{k+1} gebruiken we weer (3.4);

2) als $\tau_k \leq 10\tau_{k-1}$ dan wordt τ_k gebruikt voor de te volgen integratiestap; voor de bepaling van τ_{k+1} gebruiken we daarna (3.4');

3) nadat we (3.4') gebruikt hebben voor de bepaling van τ_k, gebruiken we (3.4") voor de bepaling van τ_{k+l}, l = 1, 2, ...

In geval 3) wordt de stapgrootte nog onderworpen aan de conditie

$$\frac{1}{2}\,\tau_{k-1} \leq \tau_k \leq \text{alfa} * \tau_{k-1}.$$

Verder wordt τ_k steeds onderworpen aan de stabiliteitsvoorwaarde

$$\tau_k \leq \frac{\beta(n)}{\sigma(D_k)}. \tag{3.5}$$

Het kan gebeuren dat $\frac{\beta(n)}{\sigma(D_k)}$ kleiner is dan $\varepsilon * t$, waarbij ε de relatieve precisie is van de rekenautomaat die we gebruiken ($\varepsilon \sim 10^{-12}$ voor de EL X8). In dat geval geldt $t_k = t_{k+1}$ en zijn dus verder berekeningen praktisch nutteloos zodat we de procedure modified runge kutta beëindigen.

3.8. De procedure normfunction

Als de parameter nrm de actuele waarde 1 heeft, krijgt normfunction de waarde van de maximumnorm van het array w. Normfunction krijgt de waarde van de Euclidische norm van w als nrm = 2. Voor andere actuele waarden van nrm geldt normfunction = 0.

3.9. De procedures vecvec, sum en rnksolelm

Deze procedure sum staat beschreven in [7].
Voor vecvec en rnksolelm verwijzen we naar [4].

4. Testresultaten

In deze sectie bespreken we een drietal problemen waarmee we modified runge kutta hebben getest, te weten de differentiaalvergelijking van Rayleigh, een hyperbolische differentiaalvergelijking en een drie-lichamen probleem in twee dimensies.

4.1. De differentiaalvergelijking van Rayleigh

Deze vergelijking heeft de volgende gedaante

$$\ddot{x} - \mu\dot{x} + \frac{1}{3}\mu(\dot{x})^3 + x = 0, \qquad x = x(t),\ \mu = \text{const.} \tag{4.1}$$

Differentiëren we (4.1) naar t dan blijkt dat $\dot{x}$ aan de vergelijking van Van der Pol voldoet

$$\dddot{x} - \mu(1-(\dot{x})^2)\ddot{x} + \dot{x} = 0.$$

We schrijven (4.1) op de volgende manier als een stelsel

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \mu(1-\frac{1}{3}(x_2)^2)x_2 - x_1, \end{cases} \tag{4.2}$$

waarin x_1 dus de oplossing van Rayleigh's vergelijking is en x_2 de oplossing van de vergelijking van Van der Pol.

We nemen voor μ de waarde 10 en kiezen verder als startwaarden

$$x_1(0) = -\frac{20}{3}, \quad x_2(0) = 2,$$

zodat de startwaarde van $\ddot{x}$ (uit (4.1)) 0 is.
De integratie van (4.2) werd voortgezet tot het tweede nulpunt van $\ddot{x}$, t = 18.86305053. Met een 5^e orde nauwkeurige Runge-Kutta methode (RK 4n uit [5]) verkregen we op dit punt

$$x = -7.09931864,\ \dot{x} = 2.01428536,\ \ddot{x} = 0. \tag{4.3}$$

Uitgaande van deze waarden hebben we modified runge kutta vergeleken met de procedures Runge 2n en Runge 3n uit [6]. Deze procedures zijn gebaseerd op 2^e respectievelijk 3^e orde nauwkeurige Runge-Kutta methoden. De resultaten volgen in de tabellen 4.1. en 4.2.
Hierin geven we onder ε_x, $\varepsilon_{\dot{x}}$ en $\varepsilon_{\ddot{x}}$ de afwijkingen van de behaalde waarden van x, $\dot{x}$ en $\ddot{x}$ met de overeenkomstige waarden uit (4.3) weer.
Onder een functie-evaluatie (f.e.) verstaan we een evaluatie van H(x,t), in dit geval dus het rechterlid van (4.2).
Uiteraard berekenden we $\ddot{x}$ steeds met behulp van (4.1).

Tabel 4.1.

Resultaten met 2^e orde Runge-Kutta methoden

methode	aantal f.e.per stap	totaal aantal f.e.	aantal stappen	$\lvert\varepsilon_x\rvert$	$\lvert\varepsilon_{\dot{x}}\rvert$	$\lvert\varepsilon_{\ddot{x}}\rvert$	locale fout-schatting
Runge 2n	2	2036	938	2.0×10^{-3}	3.3×10^{-5}	1.0×10^{-3}	l.c.t.
-	2	2414	1128	1.1×10^{-3}	2.2×10^{-5}	4.7×10^{-4}	l.c.t.
-	2	2778	1308	6.8×10^{-4}	1.6×10^{-5}	2.1×10^{-4}	l.c.t.
-	2	3362	1598	3.1×10^{-4}	1.0×10^{-5}	8.7×10^{-6}	l.c.t.
modified runge kutta	4	1564	391	1.4×10^{-3}	3.7×10^{-5}	3.0×10^{-4}	e.v.t.
-	3	2364	788	2.6×10^{-4}	3.1×10^{-5}	1.2×10^{-3}	l.c.t.
-	4	2604	651	4.6×10^{-4}	9.8×10^{-6}	1.6×10^{-4}	e.v.t.
-	3	3345	1115	1.3×10^{-4}	2.5×10^{-6}	5.6×10^{-5}	l.c.t.

Tabel 4.2.

Resultaten met 3^e orde Runge-Kutta methoden

methode	aantal f.e.per stap	totaal aantal f.e.	aantal stappen	$\lvert\varepsilon_x\rvert$	$\lvert\varepsilon_{\dot{x}}\rvert$	$\lvert\varepsilon_{\ddot{x}}\rvert$	locale fout-schatting
Runge 3n	3	2092	583	4.7×10^{-5}	7.1×10^{-7}	2.6×10^{-5}	l.c.t.
-	3	2233	634	3.7×10^{-5}	5.5×10^{-7}	2.0×10^{-5}	l.c.t
-	3	3238	953	9.5×10^{-6}	1.5×10^{-7}	5.1×10^{-6}	l.c.t.
modified runge kutta	4	2156	539	2.7×10^{-5}	1.4×10^{-6}	1.4×10^{-5}	l.c.t.
-	5	2215	443	1.8×10^{-5}	1×10^{-8}	1.8×10^{-5}	l.c.t.
-	6	3156	526	8.4×10^{-6}	3×10^{-8}	7.6×10^{-6}	l.c.t.

Uit bovenstaande tabellen blijkt dat de Runge procedures en modified runge kutta over het algemeen even nauwkeurig zijn.
Verder valt het op dat modified runge kutta in alle beschouwde gevallen met een - soms aanzienlijk - minder aantal stappen integreert.
Enerzijds komt dit doordat modified runge kutta soms een schatting van de e.v.t. kan maken waarmee in de regel een grotere stapgrootte berekend wordt dan met een l.c.t.
Anderzijds is dit het gevolg van de gebruikte extrapolatie formule voor de staplengte; de Runge procedures gebruiken hiervoor een twee-punts formule terwijl modified runge kutta volgens (3.4") een drie-punts formule gebruikt.

Tot slot van dit testvoorbeeld nog een opmerking over het stelsel (4.2).
In [6] wordt de vergelijking van Van der Pol opgelost met behulp van het stelsel

$$(4.4)\qquad \begin{cases} \dot{y}_1 = y_2, \\ \dot{y}_2 = 10(1-y_1^2)y_2 - y_1, \end{cases}$$

waarin y_1 de oplossing voorstelt.

In [6] wordt (4.4) onder andere opgelost met Runge 2n en Runge 3n. De waarde van μ, de beginvoorwaarden en de eindwaarde van t zijn hetzelfde als in ons testprobleem. Een vergelijking van tabel 4.1., 4.2. met tabel II en III uit [6] leert echter dat de Runge procedures minder moeite hebben met het oplossen van stelsel (4.2) dan met het stelsel (4.4).
Wat Runge 2n betreft zijn voor het oplossen van (4.4.) ongeveer 6× zoveel evaluaties nodig als voor het oplossen van (4.2).
Voor Runge 3n scheelt dit een factor 2.
Blijkbaar heeft (4.2) een iets minder niet-lineair karakter dan (4.4).

Voor de volledigheid geven we nu nog de procedure derivative en de aanroep van modified runge kutta waarmee de differentiaalvergelijking van Rayleigh werd opgelost.

```
procedure derivative (x,v); real x; array v;
begin real v1;
   v1  := v[2];
   v[2]:= muu * v1 * (1-v1*v1/3) - v[1];
   v[1]:= v1
end derivative;
```

De variabele muu heeft hierin de waarde 10.

```
modified runge kutta (t, s0, 1, 2, u, specrad, i, derivative,
                      k, data. 1.5, 2, aeta, reta, eta, rho,
                      output);
```

De variabele s0 heeft de eindwaarde, 18.86305053, van het integratie interval.

specrad is de volgende type procedure

```
real procedure specrad;
begin real y, s;
   s:= muu * (1-u[2]*u[2]); y:= s*s-4;
   if s < 0 then s:= if y > 0 then (s-sqrt(y))/2 else - sqrt(s*s-y)/2
            else s:= -1;
   specrad:= -s
end specrad;
```

Specrad krijgt zo bij een aanroep de waarde $\max(|\lambda_1|,|\lambda_2|)$ of, als Re $\lambda_{1,2} > 0$ de waarde 1. Hierin zijn λ_1 en λ_2 de wortels van de karakteristieke vergelijking van de Jacobiaan van (4.2)

$$\lambda^2 - \mu(1-(u[2])^2)\lambda + 1.$$

Echter, de zo door stabiliteitsoverwegingen voorgeschreven maximale tijdstap, data[0]/specrad = τ_{stab}, bleek in alle beschouwde testgevallen van de differentiaalvergelijking van Rayleigh veel groter te zijn dan de door nauwkeurigheid voorgeschreven tijdstap. Om werk te besparen hadden we dus steeds voor de actuele waarde van de parameter sigma bijvoorbeeld de waarde 1 kunnen nemen.

4.2. De differentiaalvergelijking $U_t = \frac{1}{2}U_x$

Om aan te tonen dat modified runge kutta ook gebruikt kan worden voor het oplossen van partiële differentiaalvergelijkingen, kozen we als tweede testvoorbeeld het volgende Cauchy probleem

$$(4.5)\qquad \begin{cases} U_t = \frac{1}{2}U_x, & -\infty \leq x \leq \infty, \quad 0 \leq t \leq \infty, \\ U = e^{-x^2}, & -\infty \leq x \leq \infty, \quad t = 0. \end{cases}$$

Als we de plaatsvariabele x vervangen door de discrete variabele $j\xi$, $j = 0, \pm1, \pm2, \ldots$, waarin we ξ als maaswijdte nemen, kunnen we (4.5) benaderen door een oneindig stelsel gewone differentiaalvergelijkingen van de vorm

$$(4.6) \qquad \frac{dU}{dt} = \frac{1}{4\xi}(X_+ - X_-)U.$$

Hierin is dus U een vector met oneindig veel componenten die corresponderen met de roosterpunten $j\xi$. X_+ en X_- zijn verschuivingsoperatoren met betrekking tot de index j van de componenten van U. De operator $(X_+ - X_-)$ kan gerepresenteerd worden door de volgende matrix van oneindige orde

$$(4.7) \quad D = \frac{1}{4\xi} \begin{pmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & & & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ \cdots & \cdots & 0 & -1 & 0 & 1 & 0 & \cdots & \cdots & \cdots & \\ & \cdots & \cdots & 0 & -1 & 0 & 1 & 0 & \cdots & \cdots & \cdots \\ & & \cdots & 0 & -1 & 0 & 1 & 0 & \cdots & \cdots & \cdots \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

Het is gemakkelijk na te gaan dat D zuiver imaginaire eigenwaarden heeft van de vorm

$$\lambda = \frac{i}{2\xi} \sin \omega\xi \text{ met } \omega \text{ een willekeurig reëel getal.}$$

De spectraalradius van D is dus

$$\sigma(D) = \frac{1}{2\xi}.$$

In [2] (sectie 5.1) worden stabiliteitspolynomen gegeven voor de integratie van differentiaalvergelijkingen van het type (4.6). Van deze polynomen kunnen we

$$(4.8) \qquad P_3(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{4}z^3, \qquad \beta(3) = 2$$

en

$$(4.9) \qquad P_4(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, \qquad \beta(4) = 2\sqrt{2},$$

gebruiken voor 2^e en 3^e orde exacte Runge-Kutta schema's in modified runge kutta. Met deze schema's kunnen we de analytische oplossing van (4.5), $e^{-(x+\frac{1}{2}t)^2}$, benaderen tot op termen die respectievelijk

$$(4.10)\qquad \begin{cases} O(\tau^3) + O(\tau\xi^2) & \text{(schema geïnduceerd door (4.8))} \\ \text{en} & \\ O(\tau^4) + O(\tau\xi^2) & \text{(schema geïnduceerd door (4.9))} \end{cases}$$

bedragen. (4.10) bestaat uit een afbreekfout en discretisatiefout ($O(\tau\xi^2)$). Aangezien de stabiliteitsconditie in de gevallen (4.8) en (4.9) van de vorm

$$(4.11)\qquad \tau_{stab} \leq \frac{\beta(n)}{\sigma(D)} = 2\beta(n)\xi$$

is en dus een maximale tijdstap van de orde ξ toelaat, maken we op zijn minst een totale benaderingsfout van de orde ξ^3.

Stel dat we de oplossing van (4.6) willen weten in het punt $(j\xi, t_k)$, dan hoeven we slechts de waarden van de oplossing te weten in de punten $(j\xi, t_{k-1})$, $((j\pm1)\xi, t_{k-1}), \ldots, ((j\pm n)\xi, t_{k-1})$. Hierin is n weer het aantal functie-evaluaties per integratiestap. Als we nu de oplossing in bijvoorbeeld J punten willen weten en we hebben een ruwe schatting, zeg K, van het aantal integratiestappen dat daarvoor nodig is, dan moeten we beginnen met een stelsel van $J + 2Kn$ vergelijkingen. Omdat echter gedurende het integratieproces het aantal relevante vergelijkingen minder wordt, is het efficiënt om de indices m0 en m te variëren. Stel dat U uit (4.6) bekend is in de punten $x = j\xi$, $j = g0, g0+1, \ldots, g$.

De indices m0 en m geven we dan de waarden

$$(4.12)\qquad \begin{cases} m0 = g0+i+1, & k = 0, \\ m0 = g0+(k-1)n+i+1, & k > 0, \\ m = g-i-1, & k = 0, \\ m = g-(k-1)n-i-1, & k > 0. \end{cases}$$

Hierin houden we rekening met het feit dat de parameter i in modified runge kutta de waarde -1 kan aannemen.

We geven nu achtereenvolgens de procedures m0 en derivative.

```
integer procedure m0;
begin integer decr;
   decr:= if k = 0 then i + 1 else (k-1) * data[-2] + i + 1;
   m0  := g0 + decr;
   m   := g - decr
end;

procedure derivative (x,v); real x; array v;
begin integer j; real vjm1, vj;
   vjm1:= v[m0-1];
   for j:= m0 step 1 until m do
   begin vj:= v[j];
      v[j]:= (v[j+1]-vjm1)/4/ksi;
      vjm1:= vj
   end
end;
```

De maaswijdte ksi moet gespecificeerd worden voor een aanroep van modified runge kutta.
We stelden ons tot doel de oplossing van (4.5) in het gebied

$$R: [-.06 \leq x \leq .06] \times [0 \leq t \leq .6]$$

te berekenen.
Omdat we een maaswijdte $\xi = .003$ kozen moest de oplossing dus voor $t = .6$ op minstens 40 roosterpunten bekend zijn. Ongelijkheid (4.11) schrijft een maximale stap voor van .012 en $.012\sqrt{2}$ voor de polynomen (4.8) respectievelijk (4.9). Gebruiken we voor ons Runge-Kutta schema polynoom (4.8) dan moeten we dus starten met minstens

$$40 + 2 \times \frac{.6}{.012} \times n = 340 \text{ vergelijkingen.}$$

Het gebruik van (4.9) schrijft een start met minstens 323 vergelijkingen voor.
Als de nauwkeurigheidsconditie echter tot kleinere stappen zou nopen dan (4.11), zullen we meer vergelijkingen nodig hebben om te starten.
Daarom kozen we de veilige grenzen

$$g = -g0 = 200$$

voor beide polynomen.
De actuele aanroep van modified runge kutta werd

```
modified runge kutta (t, if k > 60 then t else .6, m0, m,
                                u, 1/2/ksi, i, derivative, k, data,
                                1.5, 2, aeta, reta, rho, output);
```

In tabel 4.3. geven we nu enige resultaten behaald met een Runge-Kutta schema gebaseerd op polynoom (4.8). We integreerden met vaste stappen τ_{stab} volgens (4.11).
De getallen in de kolommen eps hebben de volgende betekenis

$$eps(t,x) = e^{-(x+\frac{1}{2}t)^2} - u(t,x),$$

waarin u(t,x) de numerieke oplossing is.

Zoals te verwachten was (uitgaande van de analytische oplossing $e^{-(x+\frac{1}{2}t)^2}$) zien we in de tabel dat voor $x + \frac{1}{2}t$ = constant ongeveer dezelfde waarden voor u worden aangenomen, bijvoorbeeld
$u(.6,-.06) \approx u(.48,0)$.
Om de betrouwbaarheid van de oplossingsmethode verder te toetsen zullen we een analyse geven van de analytische fout $\overline{eps}(.012,-.06)$.
Uitgaande van de oplossing $U(t,x) = e^{-(x+\frac{1}{2}t)^2}$ vinden we

$$U_{ttt} = U \times (x+\tfrac{1}{2}t) \times [\tfrac{3}{2} - (x+\tfrac{1}{2}t)^2].$$

Tabel 4.3.

Enige numerieke waarden van de oplossing van $U_t = \frac{1}{2}U_x$

t	u(t,-.06)	eps(t,-.06)	u(t,0)	eps(t,0)	u(t,.06)	eps(t,.06)
.012	.997088	$.1932\times10^{-7}$	.999964	0	.995653	$-.1932\times10^{-7}$
.060	.999100	$.5819\times10^{-7}$	.999100	$.3884\times10^{-7}$	.991933	$-.1345\times10^{-6}$
.120	1.000000	$.1944\times10^{-7}$	.996407	$-.1741\times10^{-6}$	.985704	$-.3614\times10^{-6}$
.180	.999101	$-.1165\times10^{-6}$	.991933	$-.4034\times10^{-6}$	.977752	$-.6759\times10^{-6}$
.240	.996407	$-.3482\times10^{-6}$	.985704	$-.7228\times10^{-6}$	.968120	$-.1072\times10^{-5}$
.300	.991933	$-.6724\times10^{-6}$	.977752	$-.1127\times10^{-5}$	.956860	$-.1541\times10^{-5}$
.360	.985704	$-.1084\times10^{-5}$	.968121	$-.1608\times10^{-5}$	.944030	$-.2075\times10^{-5}$
.420	.977753	$-.1577\times10^{-5}$	.956860	$-.2157\times10^{-5}$	.929696	$-.2662\times10^{-5}$
.480	.968121	$-.2143\times10^{-5}$	.944030	$-.2766\times10^{-5}$	.913934	$-.3293\times10^{-5}$
.540	.956861	$-.2774\times10^{-5}$	.929697	$-.3423\times10^{-5}$	.896824	$-.3955\times10^{-5}$
.600	.944031	$-.3458\times10^{-5}$	.913935	$-.4116\times10^{-5}$	.878451	$-.4636\times10^{-5}$

Polynoom (4.8) geeft de volgende benadering van de afbreekfout

$$(4.13) \qquad (\frac{1}{6} - \frac{1}{4})\tau^3 U_{ttt} = -\frac{1}{12}\tau^3 U_{ttt}.$$

De discretisatiefout, die we maken in een integratiestap, verkrijgen we door ontwikkeling van de operator in het rechterlid van (4.6)

$$(4.14) \qquad -\tau\xi^2 * \frac{1}{12} U_{xxx} = -\frac{8}{12}\tau\xi^2 U_{ttt}.$$

De totale fout $\overline{\text{eps}}$ bij een integratiestap $\tau = 2\beta(n)\xi = 4\xi$ zal bij benadering gelijk zijn aan de som van (4.13) en (4.14)

$$\overline{\text{eps}} = (-\frac{1}{12}(4\xi)^3 - \frac{8}{12}(4\xi)\xi^2)U_{ttt} = -8\xi^3 U_{ttt}.$$

Met behulp van de waarde van $U_{ttt}(0,-.06)$ krijgen we zo

$$\overline{\text{eps}}(.012,-.06) = -8(.003)^3 * U'(0,-.06) * (-.06+0)[-(-.06+0)^2+\frac{3}{2}]$$

$$= .1932 * 10^{-7},$$

geheel overeenkomstig de waarde van eps(.012,-.06) in tabel 4.3.

4.3. Een drie-lichamen probleem in twee dimensies

Om de noodzaak en het voordeel van een goede stapkeuze-strategie aan te tonen kozen we als derde testvoorbeeld een drie-lichamen probleem, waarbij de stapgrootte over een bepaald interval sterk afhankelijk bleek te zijn van de beginvoorwaarden.
Het probleem wordt gegeven door de volgende stelsels (versnellingen)

$$(4.15) \qquad \ddot{x}_i = \sum_{\substack{j=1 \\ i\neq j}}^{3} -\frac{x_i-x_j}{r_{ij}^3}, \qquad \ddot{y}_i = \sum_{\substack{j=1 \\ i\neq j}}^{3} -\frac{y_i-y_j}{r_{ij}^3},$$

$$(r_{ij} = \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2}), \quad i = 1, 2, 3,$$

met beginvoorwaarden

$$(4.16) \qquad \begin{cases} x_1 = -x_2 = -1, \quad x_3 = 1.5, & \dot{x}_1 = \dot{x}_2 = \dot{x}_3 = 0, \\ y_1 = y_2 = 0, \quad y_3 = 10, & \dot{y}_1 = -\dot{y}_2 = -\frac{1}{2}, \quad \dot{y}_3 = -2. \end{cases}$$

Het aldus gedefinieerde stelsel kunnen we opvatten als een wiskundig model van een dubbelster-stelsel waarlangs met hoge snelheid een derde ster beweegt. De massa's van de sterren zijn gelijk en komen overeen met één zonsmassa. Een afstandseenheid en een tijdseenheid komen overeen met respectievelijk 2.36 astronomische eenheden en 1 jaar.

We geven nu allereerst onze versie van de procedure derivative voor het stelsel (4.15)

```
procedure derivative (x,v); real x; array v;
begin integer j, l, p2, p3;
   real s, t, flx, fly, difx, dify, flxj, flyj, rad;
   p2:= p+p; p3:= p2+p;
   for l:= 1 step 1 until p do
   begin a[l]:= v[l+p2]; a[l+p]:= v[l+p3];
      v[l+p2]:= v[l+p3]:= 0
   end;
   for l:= 1 step 1 until p-1 do
   begin s:= v[l]; t:= v[l+p]; flx:= fly:= 0;
      for j:= l+1 step 1 until p do
      begin difx:= v[j]-s; dify:= v[j+p]-t;
         rad:= difx*difx+dify*dify; rad:= 1/rad/sqrt(rad);
         flxj:= difx*rad; flyj:= dify*rad;
         flx:= flx+flxj; fly:= fly+flyj;
         v[j+p2]:= v[j+p2]-flxj; v[j+p3]:= v[j+p3]-flyj
      end;
      v[l]:= a[l]; v[l+p2]:= v[l+p2]+flx;
      v[l+p]:= a[l+p]; v[l+p3]:= v[l+p3]+fly
   end;
   v[p]:= a[p]; v[p2]:= a[p2]
end derivative;
```

De variabele p heeft hierin de waarde 3 (het aantal lichamen), a[1:2*p] is een array om tijdelijk de waarden van v[2*p+1], ..., v[4*p] op te slaan.

Als actuele aanroep van modified runge kutta namen we

```
modified runge kutta (t, tend, 1, p4, u, 1, i, derivative, k,
                      data, 2, 2, aeta, reta, eta, rho,
                      output);
```

p4 heeft hierin de waarde 4 * p (=12). Het array u bevat de start-waarden van, achtereenvolgens

(4.17) $x_1, x_2, x_3, y_1, y_2, y_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{y}_1, \dot{y}_2, \dot{y}_3$.

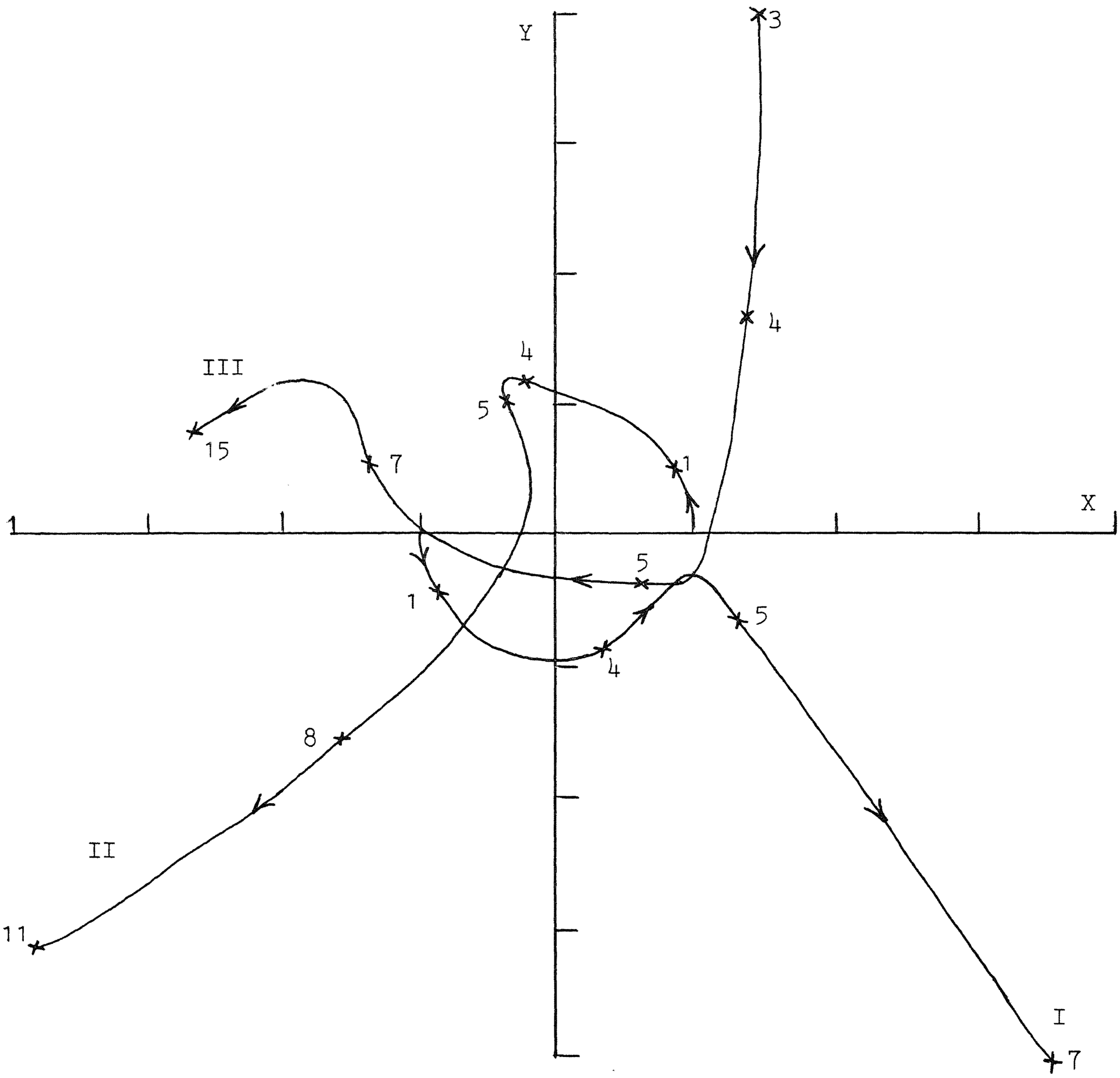

fig. 1.

Drie-lichamen probleem in twee dimensies

Het array data vulden we met de volgende waarden:

$$\text{data}[-3] = \text{data}[-2] = 4,\ \text{data}[-1] = 3,\ \text{data}[0] = 6,$$

$$\text{data}[1] = 1,\ \text{data}[2] = \frac{1}{2},\ \text{data}[3] = \frac{1}{6},\ \text{data}[4] = .18455702\times 10^{-1}.$$

De waarde 1 voor de spectraalradius σ verkregen we door een afschatting met behulp van Gerschgorin's theorema.
De parameter tend kreeg als eindwaarde 25.
Figuur 1 toont een beeld van de verkregen oplossing. De beginposities zijn, zoals gegeven in (4.16)

(-1,0) voor lichaam I,

(1,0) voor lichaam II en

(1.5,10) voor lichaam III.

Op de banen hebben we de posities van de lichamen op bepaalde tijdstippen gemarkeerd met een ✕.

In de volgende tabel geven we het aantal integratiestappen per tijdsinterval [t,t+1], t = 0, 1, ..., 12.
Het grote aantal integratiestappen over het interval [4,5] wordt veroorzaakt door de sterke gravitatiekrachten gedurende dat tijdsinterval. Zoals uit figuur 1 blijkt, bevinden de lichamen I en III zich op dat moment dicht bij elkaar.

De kleinste integratiestap werd genomen op het interval $4.780 < t < 4.785$:

$$\tau_{min} \approx 4.8 * 10^{-4}.$$

Voor $t > 22$ werd met de grootste stappen geïntegreerd:

$$\tau_{max} = 1,$$

zodat over het gehele interval $0 \leq t \leq 25$ de staplengte varieerde met een factor

$$\frac{\tau_{max}}{\tau_{min}} \approx 2080.$$

Tabel 4.4.

Aantal integratiestappen per tijdsinterval

interval	aantal integratiestappen
[0,1]	9
[1,2]	6
[2,3]	7
[3,4]	11
[4,5]	152
[5,6]	28
[6,7]	25
[7,8]	11
[8,9]	6
[9,10]	4
[10,11]	3
[11,12]	2
[12,13]	2

Referenties

[1] P.J. van der Houwen — Stabilized Runge-Kutta methods with limited storage requirements (TW124) Mathematisch Centrum.

[2] P.J. van der Houwen — One step methods for linear initial value problems I (TW119) Mathematisch Centrum.

[3] P.J. van der Houwen — One step methods for linear initial value problems II (TW122) Mathematisch Centrum.

[4] T.J. Dekker — Algol 60 procedures in numerical algebra Part 1 Mathematical Centre Tracts 22 (1970)

[5] J.A. Zonneveld — Automatic numerical integration Mathematical Centre Tracts 8 (1970)

[6] W.E. Baanstra — Methoden voor het oplossen van beginwaardeproblemen (NR12) Mathematisch Centrum.

[7] D. Grune — Handleiding Milli-systeem voor de EL X8 Mathematisch Centrum.