

3. RANDWAARDEPROBLEMEN

3.1. Oplossen van tweepunts randwaardeproblemen

door P.W. Hemker

(Mathematisch Centrum)

en

J.P. Roos

(Corporate Research, Akzo Arnhem)

### 3.1.1. Inleiding

Voor tweepunts randwaardeproblemen bestaat weinig programmatuur die eenvoudig toegankelijk is. In de programmatheken ACCULIB en IMSL vinden we geen programma's die direkt in staat zijn een tweepunts randwaardeprobleem op te lossen. In de NAG-programmatheek vinden we 2 routines en in de NUMAL vinden we 4 routines voor bepaalde klassen van tweepunts randwaardeproblemen. Bovendien vinden we in NUMAL een routine voor het schatten van onbekende parameters, die ook voor tweepunts randwaardeproblemen gebruikt kan worden.

De oorzaak van dit gebrek aan algemene routines voor randwaardeproblemen kan wellicht gevonden worden in het feit dat programma's voor deze problemen praktisch altijd efficiënter worden wanneer ze zich beperken tot een deelklasse van problemen. Dit houdt in dat men snel geneigd zal zijn voor ieder probleem afzonderlijk een programma te schrijven dat juist dat éne probleem efficiënt oplost. We vinden dan ook, naast de routines die speciaal bestemd zijn voor randwaardeproblemen, in de meeste programmatheken een arsenaal van routines voor het oplossen van ijle lineaire stelsels. Deze routines maken het mogelijk om randwaardeproblemen op te lossen nadat de gebruiker ze zelf heeft gediscretiseerd. Voor de tweepunts randwaardeproblemen zijn hier vooral nuttig de routines die lineaire stelsels oplossen waarvan de matrix een bandstructuur heeft.

Deze noodzaak tot het schrijven van ad hoc programma's zal echter verdwijnen als er toch voldoende efficiënte programma's voor algemene problemen gemaakt worden. Naast de programmatuur die in de programmatheken beschikbaar is, zijn er op verschillende plaatsen programma's in voorbereiding. We willen hier in het bijzonder vermelden een programma van Bulirsch en Stoer, gebaseerd op de multiple-shooting techniek (de definitieve publicatie is vornsog onbekend) en programma's door Keller en Pereyra die gebaseerd zijn op een discretiserings-methode.

### 3.1.2. Oplossings methoden

Vele studies zijn verricht op het gebied van het numeriek oplossen van tweepunts randwaardeproblemen en nog steeds worden nieuwe en betere algoritmen ontwikkeld. Hoewel het onmogelijk is hier een overzicht te geven van de bestaande theorieën en methoden, is het toch van belang de

belangrijkste principes kort uiteen te zetten.

Een tweepunts randwaardeprobleem bestaat uit een (stelsel) differentiaalvergelijking(en) die gedefinieerd is (zijn) op een interval  $[a,b]$ ; bovendien zijn condities van de functie gegeven in  $a$  en  $b$ . Voor het numeriek oplossen van deze problemen onderscheiden we twee klassen van methoden: de *schietmethoden* en de *discretiseringsmethoden*. Voor de beide basisprincipes zijn veel varianten mogelijk.

### Schietmethoden

Schietmethoden reduceren het probleem tot een serie beginwaardeproblemen. De differentiaalvergelijking wordt geschreven als een stelsel van  $n$  eerste orde vergelijkingen en aan de rand (zeg  $x = a$ ) wordt een aantal beginwaarden gekozen die in overeenstemming zijn met de randcondities voor  $x = a$ . De ontbrekende beginvoorwaarden bij  $x = a$  worden geïntroduceerd als onbekende parameters.

Met verschillende waarden voor deze parameters wordt het beginwaardeprobleem geïntegreerd van  $a$  naar  $b$ . Aan de hand van de verkregen oplossing in  $b$  worden de parameters, in een iteratief proces, zodanig bepaald dat het beginwaardeprobleem aan de condities van het randwaardeprobleem voldoet. Verschillende varianten op deze schietmethode zijn mogelijk:

1. integratie van het beginwaardeprobleem van  $b$  naar  $a$  i.p.v. van  $a$  naar  $b$ .
2. integratie van  $a$  naar  $r$ ,  $a < r < b$ , en van  $b$  naar  $r$ . In deze variant worden aan beide randen onbekende parameters geïntroduceerd. Deze parameters worden nu zó bepaald dat de oplossingen op de intervallen  $(a,r)$  en  $(r,b)$  op elkaar aansluiten in  $x = r$  (continuïteit van de oplossing).
3. multiple-shooting.

Hierbij wordt het interval  $[a,b]$  verdeeld in meer deelintervallen

$$[x_{i-1}, x_i], \quad a = x_0 < x_1 < \dots < x_N = b.$$

Op ieder deelinterval wordt een beginwaardeprobleem opgelost. De parameters voor de onbekende beginvoorwaarden worden nu zodanig bepaald dat de uiteindelijke oplossing continu is in alle punten  $x_i$ ,  $1 \leq i \leq N - 1$ , en zodat aan de randvoorwaarden wordt voldaan.

De belangrijkste moeilijkheid bij het oplossen van tweepunts randwaardeproblemen met een schietmethode is de mogelijk grote instabiliteit van een probleem. We illustreren dit met het volgende, eenvoudige probleem

$$y'' - (p+q)y' + pq y = s, \quad a \leq x \leq b,$$

$$y(a) = \alpha, \quad y(b) = \beta.$$

Dit probleem is stabiel wanneer  $pq < 0$ ; d.w.z. als  $pq < 0$  dan veroorzaakt een kleine verstoring in de gegevens  $s, \alpha$  of  $\beta$  ook een kleine storing in de oplossing  $y$ . Het beginwaardeprobleem dat wordt opgelost wanneer een schietmethode wordt toegepast heeft een Jacobiaan met eigenwaarden  $p$  en  $q$  wanneer het in voorwaartse richting wordt geïntegreerd of met eigenwaarden  $-p$  en  $-q$  als het in achterwaartse richting wordt geïntegreerd. Voor een stabiel tweepunts randwaardeprobleem verschillen  $p$  en  $q$  van teken. Het beginwaardeprobleem is instabiel. Als voor een eigenwaarde (zeg  $p$ ) bovendien  $|p|$  groot is, dan is het beginwaardeprobleem stijf ( $p < 0$ ) of sterk instabiel ( $p > 0$ ). In beide gevallen is het berekenen van de onbekende parameters een slecht geconditioneerd probleem.

#### Discreteriseringsmethoden

Discretiseringsmethoden gaan uit van een partitie van het interval  $[a, b]$ , d.i. een (groot) aantal punten  $x_i$  zodat  $a = x_0 < x_1 < \dots < x_n = b$ . De oplossing wordt nu berekend voor deze van te voren gegeven  $x$ -waarden. Hiertoe worden de differentiaaloperator en het rechterlid van de vergelijking gediscretiseerd. Voor discretisatie behoeft de vergelijking niet gereduceerd te worden tot een stelsel eerste orde vergelijkingen. De discretisering kan op vele verschillende manieren plaatsvinden. Het uiteindelijke resultaat is echter dat de differentiaalvergelijking en de randcondities worden teruggebracht tot een (groot) stelsel algebraïsche vergelijkingen. De oplossing van dit stelsel levert de waarden van de gevraagde functie op de te voren gegeven partitie.

De belangrijkste varianten van de discretiseringsmethoden zijn de differentiemethoden en de globale methoden.

Differentiemethoden zijn het eenvoudigst. Hierbij worden de differentiaaloperatoren direkt vervangen door differentieoperatoren. De differentiemethoden worden voornamelijk toegepast voor een lage orde van nauwkeurigheid en voor uniforme partities.

Globale methoden baseren de discretisering op de keuze van een eindig aantal basisfuncties  $\{\phi_i\}$  in de lineaire ruimte van alle mogelijke oplossingsfuncties. Een numerieke oplossing

$$y_h(x) = \sum_{j=0}^M a_j \phi_j(x)$$

wordt nu bepaald zodat  $y_h(x)$ , aan de hand van zekere criteria, zo goed mogelijk aan de differentiaalvergelijking en aan de randcondities voldoet. Tot deze groep van methoden behoren o.a. de *collocatiemethoden*, de *Galerkin-methoden* en de *kleinste-kwadratenmethode*.

Globale methoden kunnen eenvoudig op een niet-uniforme partitie worden toegepast en de constructie van deze methoden met een hoge orde van nauwkeurigheid is eenvoudiger dan voor differentiemethoden.

### 3.1.3. Overzicht van de beschikbare programmatuur

In de programmatheken NAG en NUMAL zijn de volgende routines beschikbaar voor het oplossen van tweepunts randwaardeproblemen.

#### 1. DO2ADA/F (NAG)

Deze routine lost een tweepunts randwaardeprobleem op voor een stelsel gewone differentiaalvergelijkingen over een interval  $[a,b]$

$$(3.1.3.1) \quad \frac{dy_i}{dx} = f_i(x, y_1, \dots, y_N), \quad i = 1, 2, \dots, N.$$

De procedure gebruikt de bovengenoemde variant no. 2 van de schietmethode. De gebruiker specificceert een getal  $r$ ,  $a < r < b$ , waarna beginwaardeproblemen worden opgelost van  $a$  naar  $r$  en van  $b$  naar  $r$ . Naast de  $N$  vergelijkingen (3.1.3.1) moet de gebruiker  $N$  randwaarden opgeven, sommige voor  $x = a$ , de andere voor  $x = b$ . Bovendien moet de gebruiker een beginschatting geven voor de ontbrekende randwaarden van  $y$ .

De procedure levert na afloop de berekende waarden van  $y$  voor  $x = a$  en  $x = b$  en, als daarom gevraagd wordt, de berekende waarden van  $y$  op een equidistante partitie van  $[a,b]$ . De beginwaardeproblemen worden in DO2ADA/F opgelost met een Runge-Kutta-Merson methode, de onbekende parameters worden bepaald met een vorm van Newton-iteratie.

#### 2. DO2AGA/F (NAG)

Deze routine heeft veel overeenkomst met DO2ADA/F. Het randwaardeprobleem wordt weer opgelost met een schietmethode, variant 2 (met de

Runge-Kutta-Merson integratiemethode en Newton-iteratie). De routine is echter algemener in die zin, dat ook parameters die geen randwaarden zijn (eigenwaarden, coëfficiënten in de vergelijking etc.) bepaald kunnen worden. Het randwaardeprobleem dat kan worden opgelost luidt

$$(3.1.3.2) \left\{ \begin{array}{l} \frac{dy_i}{dx} = f_i(x, y_1, \dots, y_N, p_1, \dots, p_M), \quad a < x < b, \\ a = a(p_1, \dots, p_M), \quad b = b(p_1, \dots, p_M), \\ y(a) = y_a(p_1, \dots, p_M), \quad y(b) = y_b(p_1, \dots, p_M), \quad M \leq N. \end{array} \right.$$

De functies  $f_i$ ,  $a$ ,  $b$ ,  $y_a$  en  $y_b$  moeten door de gebruiker worden gespecificeerd. Ook het punt  $r$ ,  $a < r < b$ , waar de oplossingen van de beginwaardeproblemen aan elkaar gepast moeten worden, kan worden opgegeven afhankelijk van  $p_1, \dots, p_M$ . De procedure heeft een beginschatting voor  $\vec{p}$  nodig en levert een verbeterde waarde van  $\vec{p}$  af.

### 3. PEIDE (NUMAL)

Deze procedure berekent parameters  $p_1, \dots, p_M$  zodanig dat

$$(3.1.3.3) \quad \sum_{j=1}^J |y_i(x_j) - s_j|^2, \quad J \geq M,$$

geminimaliseerd wordt. Hierin is  $\{(x_j, s_j)\}_{j=1}^J$  een gegeven verzameling punten ("observaties") en  $y_i$  is een component van de oplossing van het beginwaardeprobleem

$$(3.1.3.4) \left\{ \begin{array}{l} \frac{dy_i}{dx} = f_i(x, y_1, y_2, \dots, y_N, p_1, \dots, p_M), \quad i = 1, \dots, N \\ y(a) = y_a(p_1, \dots, p_M). \end{array} \right.$$

Zo bepaalt PEIDE parameters in de vergelijking en in de randvoorwaarden, zodanig dat de oplossing van de vergelijking een gegeven verzameling "observaties" zo dicht mogelijk passeert.

De routine gebruikt de multiple-shooting techniek waarbij punten  $x_j$ ,  $a < x_j < b$ , gebruikt kunnen worden als partitie-punten. De beginwaardeproblemen worden opgelost met Gear's methode (geschikt voor stijve differ-

entiaalvergelijkingen). De minimalisering van het residu (3.1.3.3) gebeurt met de methode van Marquardt. Hierbij maakt de procedure gebruik van, door de gebruiker te specificeren, partiële afgeleiden:

$$\frac{\partial y_{a_i}}{\partial p_j}, \quad \frac{\partial f_i}{\partial y_k}, \quad \frac{\partial f_i}{\partial p_j} \quad i, k = 1, \dots, N; j = 1, \dots, M.$$

De procedure PEIDE kan gebruikt worden voor het oplossen van het tweepuntsrandwaardeprobleem (3.1.3.1) door te nemen

$$(3.1.3.5) \quad \begin{aligned} y_i(a) &= y_{a_i}(p_1, p_2, \dots, p_M) \\ x_j &= b, \quad s_j = y_{i_j}(b), \quad j = 1, 2, \dots, M. \end{aligned}$$

#### 4. FEMLAG, FEMLAGSYM, FEMLAGSKEW (NUMAL)

Deze procedures lossen een probleem op van de vorm

$$(3.1.3.6) \quad -(p(x)y'(x))' + r(x)y(x) = f(x), \quad a \leq x \leq b, \text{ of}$$

$$-y''(x) + q(x)y'(x) + r(x)y(x) = f(x), \quad a \leq x \leq b,$$

met de randvoorwaarden

$$c_1 y(a) + c_2 y'(a) = c_3,$$

$$c_4 y(b) + c_5 y'(b) = c_6.$$

De oplossing wordt verkregen op een, van te voren gegeven, partitie  $a = x_0 < x_1 < \dots < x_N = b$ . Er zijn geen beginschattingen voor parameters nodig. Deze procedures gebruiken een Galerkin-methode en zijn zeer nauwkeurig. De orde van nauwkeurigheid kan worden opgegeven en is  $O(h^2)$ ,  $O(h^4)$  of  $O(h^6)$ , waarin  $h$  de maaswijdte van de partitie is. De coëfficiënt  $p(x)$  moet positief zijn.

#### 5. FEMHERMSYM (NUMAL).

Deze procedure lost een 4-de orde differentiaalvergelijking op van de vorm

$$(3.1.3.7) \quad (p(x) y''(x))' - (q(x) y'(x))' + r(x) y(x) = f(x), \quad a \leq x \leq b$$

met de randvoorwaarden

$$\begin{aligned} y(a) &= c_1, \quad y'(a) = c_2, \\ y(b) &= c_3, \quad y'(b) = c_4. \end{aligned}$$

De waarden van  $y(x)$  en  $y'(x)$  op een van te voren gegeven partitie worden berekend met een Galerkin-methode. De orde van nauwkeurigheid is  $O(h^4)$ ,  $O(h^6)$  of  $O(h^8)$ .

#### 3.1.4. Het gebruik van de programmatuur

In deze sectie lichten we het gebruik van de programmatuur toe aan hand van een drietal problemen. Deze problemen zijn op verschillende manieren opgelost. De betreffende programmatuur en de bijbehorende numerieke resultaten kunnen worden gevonden in de bijlagen.

##### 3.1.4.1. Een zuiver tweepunts randwaardeprobleem

We willen de hoek berekenen waaronder een projectiel afgeschoten moet worden om een bepaalde horizontale afstand af te leggen. De beweging vergelijkingen zijn voor  $x(t)$  en  $y(t)$

$$(3.1.4.1) \quad \begin{cases} x'' = -\alpha v^2 \sin \phi \\ y'' = -g - \alpha v^2 \cos \phi \end{cases}$$

waarin

$$\begin{aligned} v &= \sqrt{(x')^2 + (y')^2} \quad (\text{de snelheid van het projectiel}) \\ \phi &= \arctan(dy/dx) \quad (\text{de hoek met de horizontale richting}) \end{aligned}$$

zodat de baan van het projectiel beschreven wordt door het stelsel

$$(3.1.4.2) \quad \begin{cases} \frac{dy}{dx} = \tan \phi \\ \frac{dv}{dx} = g \tan(\phi)/v - \alpha v \cos(\phi) \\ \frac{d\phi}{dx} = -g/v^2 \end{cases}$$



Met de constanten  $g = 9.80665$ ,  $\alpha = 0.0007$  en de randvoorwaarden

$$\begin{aligned} y &= 0, v = 150 \text{ voor } x = 0, \\ y &= 0 \text{ voor } x = 1500, \end{aligned}$$

willen we  $\phi(0)$  berekenen. We schatten de (overige) randwaarden als  $\phi(0) = 1.15$ ;  $\phi(1500) = -1.2$ ;  $v(1500) = 135$ .

### 3.1.4.2. Een parameter-schat probleem

Dit praktijkprobleem is ontleend aan de besliskunde. Om een curve  $y(x)$  te berekenen die een optimale verkoopstrategie bepaalt, moeten we een parameter  $p > 0$  bepalen, zodanig dat

$$(3.1.4.3) \quad y'(x) = 1 + p + \log(3) \exp(-n\ell y) \times \left[ y \sum_{j=0}^{n-2} \frac{1}{j!} (n\ell y)^j - \frac{1}{\ell} \sum_{j=0}^{n-1} \frac{1}{j!} (n\ell y)^j \right]$$

waarin  $\ell = \ell(x) = \frac{1}{3x+0.01}$  en  $n > 1$  een geheel getal.

Deze vergelijking geldt op het interval  $[0,1]$ ; de randvoorwaarden zijn  $y(0) = 0$ ,  $y(1) = 2$ . Omdat weliswaar  $p > 0$ , maar de orde van grootte  $p$  onbekend is, introduceren we de nieuwe parameter  $P = \log(p)$ . We bepalen de waarde van  $p$  voor  $n = 10$  en  $n = 20$ .

### 3.1.4.3. Een niet-lineaire tweede orde vergelijking

We willen het volgende probleem oplossen met een schietmethode en met een globale methode

$$(3.1.4.4) \quad \begin{cases} y'' = e^y & \text{op } [0,1], \\ y(0) = y(1) = 0. \end{cases}$$

De analytische oplossing is bekend en luidt

$$(3.1.4.5) \quad y(x) = 2 \ln\left(\frac{c}{\cos(e(x-0.5))}\right) + \ln 2,$$

$$c \approx 0.6680278475.$$

Voor de schietmethode schrijven we het probleem als

$$(3.1.4.6) \quad \begin{cases} y' = v \\ v' = e^y \end{cases},$$

$$y(0) = y(1) = 0.$$

Om het probleem op te lossen met een globale methode gebruiken we de procedure FEMLAG. Deze procedure is alleen geschikt voor lineaire problemen en daarom construeren we het iteratieve Newton-proces

$$(3.1.4.7) \quad -y_{n+1}'' + [e^{y_n}]y_{n+1} = [e^{y_n}(y_n - 1)].$$

3.1.4.4. Een probleem dat niet opgelost kon worden met de beschikbare programmatuur

Vele tweepunts randwaardeproblemen leveren toch grote moeilijkheden op wanneer we alleen beschikken over de programmatuur die in sectie 3.1.3 genoemd is. Zo bleek het niet mogelijk om met een van de genoemde routines het volgende probleem op te lossen

$$(3.1.4.8) \quad \begin{cases} \varepsilon p y'' = g(y-z) \\ p z'' = -g(y-z) \end{cases} \quad \text{op } [0,1]$$

$$\begin{aligned} y'(0) &= 0, \quad z(0) = 0, \\ y(1) &= 1, \quad z'(1) = 0, \end{aligned}$$

waarin

$$g(x) = \exp\left(\frac{1}{3}\eta x\right) - \exp\left(-\frac{2}{3}\eta x\right)$$

$$(3.1.4.9) \quad \varepsilon = 0.143, \quad p = 0.1743, \quad \eta = 17.19.$$

Dit probleem heeft zo een duidelijk stijf karakter. Ook echter met de parameters (waarmee het probleem minder stijf is)

$$\epsilon = 0.685, p = 4.028, \eta = 17.23$$

bleken de schietmethoden te falen. We wijten dit aan de sterke niet-lineariteit van het probleem die te grote moeilijkheden veroorzaakt voor de gebruikte integratiemethoden.

### 3.1.5. De algemene routines ontwikkeld door Pereyra en zijn medewerkers

Zoals in de inleiding tot dit hoofdstuk reeds werd aangegeven, bestaan er nauwelijks algemene routines voor het oplossen van tweepunts randwaardeproblemen.

Het is bekend, dat o.a. Keller en Pereyra, elk met hun medewerkers, actief zijn op dit gebied. PEREYRA [1974] is echter de enige die een algemeen bruikbare routine (SYSSOL) gepubliceerd heeft; bovendien heeft hij reeds de beschikking over een experimentele, verbeterde versie (PASVAR), waarmee ook niet-uniforme roosters behandeld kunnen worden.

Bij zijn ontwikkeling van algemeen bruikbare routines voor het oplossen van randwaardeproblemen heeft Pereyra de bedoeling .. *to achieve the quality and high standards of the general purpose software available for initial value problems*. En ons inziens is hij daarin vrijwel geslaagd.

We zullen ons beperken tot tweepunts randwaardeproblemen van het type:

$$\frac{dy}{dx} = f(x, y)$$

$$(3.1.5.1) \quad a \leq x \leq b$$

$$A_1 y(a) + B_1 y(b) = C,$$

met

$y$  en  $f$ :  $M$ -dimensionale vektorfuncties,

$A$  en  $B$ : konstante ( $M \times M$ ) matrices,

$C$ : konstante  $M$ -vektor.

We nemen aan dat de gezochte oplossing  $y$  van (3.1.5.1) bestaat en geïsoleerd is. Het gestelde probleem is dan:

*Bereken een oplossing  $y$  die over het gehele interval  $[a,b]$  slechts met een bij voorbaat aangegeven tolerantie van de exakte oplossing  $y$  mag afwijken.*

De door Pereyra behandelde voorbeelden en enkele toepassingen van ons lijken de konklusie te wettigen dat de routines SYSSOL en vooral PASVAR het betrekkelijk automatisch en simpel oplossen van problemen als (3.1.5.1) mogelijk maken.

De structuur van SYSSOL en PASVAR is globaal als volgt. De basis is een *discretiseringsmethode* (vgl. 3.1.2), waarbij gebruik gemaakt wordt van eindige differenties.

Dit betekent o.a. dat een bepaald rooster (partitie, mesh of grid) gegeven moet zijn. Bij een gegeven rooster leidt de discretisering tot een stelsel van eindig veel *niet-lineaire vergelijkingen*. De methode van Newton wordt toegepast om dit stelsel op te lossen, vgl. PEREYRA [1968, 1974, 1975], KELLER [1974]. Op een efficiënte wijze wordt door PEREYRA [1974] gebruik gemaakt van de bandstructuur.

Na de oplossing van dit stelsel vergelijkingen is dus een *benadering* van de exakte oplossing op het gegeven rooster bekend. Maar hoe goed is deze benadering?

PEREYRA [1968] past nu de techniek van *Iterated Deferred Corrections (IDC)* toe om op het gegeven rooster een schatting van de fout te bepalen en vervolgens de benaderde oplossing (iteratief) te verbeteren.

N.B. Pereyra claimt dat IDC efficiënter is dan een techniek als Richardson extrapolatie, die b.v. door KELLER [1974] wordt toegepast.

Wanneer verder verbetering niet mogelijk of zinnig lijkt, wordt het *rooster verfijnd*.

In de gepubliceerde routine SYSSOL is het slechts mogelijk met een uniform rooster te werken; dit rooster wordt door halvering verfijnd.

In de versie PASVAR kan met een niet-uniform rooster worden gestart; verfijning van het rooster is i.h.a. niet uniform. Er wordt naar een zodanig rooster gestreefd, dat de geschatte fout van de oplossing uniform over het interval verdeeld is (dus b.v. veel roosterpunten bij grotere gradiënten): het begrip "equidistribution" wordt hier gehanteerd.

De overgangen tussen de verschillende genoemde fasen worden via een aantal strategieën geregeld. De "tuning" van deze overgangen is deels op basis van experimenten bepaald.

Afgezien van de gevallen waarin de subroutine op een abnormale wijze beëindigd wordt (b.v. onjuiste invoergegevens), leveren beide subroutines op "automatische" wijze een berekende oplossing, die in het algemeen aan de *door de gebruiker opgegeven nauwkeurigheid* voldoet.

De deklaratie van de subroutine SYSSOL is als volgt:

SUBROUTINE SYSSOL (M,N,A,B,C,A1,B1,TOL,DELEPS,X,Y,ABT,FF,JACOB,JERROR).

De parameters M,A,B,C,A1,B1,X en Y hebben dezelfde betekenis als in (3.1.5.1). N is het aantal roosterpunten (inclusief de beide randpunten) waarmee wordt begonnen. FF en JACOB zijn de door de gebruiker te definiëren subroutines voor de berekening van de rechterleden f van (3.1.5.1) en van de jacobiaan ( $\partial f/\partial y$ ) van f. TOL is de door de gebruiker op te geven absolute nauwkeurigheid.

Voor de betekenis van DELEPS en ABT en voor verdere details wordt verwezen naar PEREYRA [1974].

De subroutine is eenvoudig te gebruiken en snel uit de literatuur over te nemen.

#### Enkele voorbeelden

a. Een randwaardeprobleem uit de elektriciteitsleer.

$$\epsilon_p \frac{d^2 y}{dt^2} = g(y-z) ,$$

$$p \frac{d^2 z}{dt^2} = -g(y-z) , \quad 0 \leq t \leq 1 ,$$

$$t = 0 : \frac{dy}{dt} = 0 , z = 0 ,$$

$$t = 1 : y = 1 , \frac{dz}{dt} = 0 ,$$

$$g(x) = \exp\left(\frac{1}{3}\eta x\right) - \exp\left(-\frac{2}{3}\eta x\right) ,$$

$$\epsilon = 0.143; \quad P = 0.1743; \quad \eta = 17.19.$$

De met PASVAR c.q. SYSSOL berekende oplossing is in de volgende tabel opgenomen.

t	y	z	$\frac{dt}{dy}$	$\frac{dz}{dt}$
0.000	0.033	0.000	0.000	0.899
0.125	0.103	0.102	0.762	0.790
0.250	0.201	0.201	0.786	0.787
0.375	0.299	0.299	0.787	0.787
0.500	0.397	0.397	0.787	0.787
0.625	0.496	0.496	0.787	0.787
0.750	0.594	0.594	0.793	0.786
0.875	0.700	0.691	0.987	0.758
1.000	1.000	0.761	6.287	0.000

De grenslaagstructuur bij  $t = 0$ , en vooral bij  $t = 1$ , is duidelijk te zien. Wanneer  $\epsilon$  groter wordt gekozen gaan  $y$  en  $z$  meer uiteen; er ontstaat dan een minder uitgesproken grenslaagstructuur.

Dit probleem is met behulp van (elementair) schieten, vgl. (3.1.2) niet of nauwelijks oplosbaar.

Dit probleem is zowel met SYSSOL als met PASVAR opgelost. Bij SYSSOL was een uniforme verfijning nodig tot 156 intervallen.

Bij PASVAR met gelijke nauwkeurigheidseis, werd een niet-uniforme verfijning toegepast tot 84 intervallen, waarbij 16 kleine op het deelinterval  $[\frac{15}{16}, 1]$ ; dit geeft aan dat het rooster aangepast werd aan het karakter van de oplossing.

Een wat eenvoudiger probleem behoort bij de parameterwaarden  $\epsilon = 0.685$ ;  $p = 4.028$ ;  $\eta = 17.23$ . De oplossing is dan:

t	y	z	$\frac{dy}{dt}$	$\frac{dz}{dt}$
0.000	0.348	0.000	0.000	1.001
0.125	0.365	0.114	0.247	0.832
0.250	0.406	0.211	0.401	0.726
0.375	0.464	0.296	0.518	0.646
0.500	0.536	0.373	0.629	0.570
0.625	0.621	0.439	0.740	0.494
0.750	0.722	0.495	0.882	0.397
0.875	0.844	0.537	1.085	0.258
1.000	1.000	0.555	1.461	0.000

Reeds bij 29 intervallen werd hier de gewenste nauwkeurigheid (meer dan de gegeven cijfers) bereikt.

b. Een probleem dat voortkomt uit de berekening van de warmte- en impuls-overdracht tussen een bewegende cylinder en een stilstaande omgeving

$$t \frac{d^3 y}{dt^3} - \left(\frac{dt}{dt}\right)^2 + (y-1) \left(\frac{d^2 y}{dt^2} - \frac{1}{t} \frac{dy}{dt}\right) = 0,$$

$$t \frac{d^2 z}{dt^2} + (py+1) \frac{dz}{dt} = 0, \quad t_0 \leq t \leq t_1,$$

$$t = t_0 : y = 0, \quad \frac{dy}{dt} = t; \quad z = 1,$$

$$t = t_1 : \frac{dy}{dt} = 0, \quad z = 0,$$

$$t_0 = 0.005; \quad t_1 = 20; \quad p = 0.72.$$

De met PASVAR berekende oplossing is:

t	y	$\frac{dt}{dy}$	$\frac{d^2 y}{dt^2}$	z	$\frac{dz}{dt}$
0.005	0.000	0.005	0.835	1.000	-28.056
0.010	0.000	0.009	0.716	0.898	-13.600
0.020	0.000	0.015	0.607	0.805	- 7.002
0.102	0.003	0.052	0.342	0.576	- 1.368
0.210	0.010	0.082	0.230	0.476	- 0.666
0.06	0.124	0.163	0.032	0.254	- 0.125
2.11	0.303	0.171	-0.007	0.167	- 0.057
3.16	0.477	0.158	-0.015	0.121	- 0.034
4.21	0.635	0.141	-0.016	0.092	- 0.023
5.27	0.775	0.125	-0.015	0.072	- 0.016
20.0	1.536	0.000	-0.006	0.000	- 0.001

Dit probleem kon ook met schieten opgelost worden.

Het is interessant te zien dat PASVAR een duidelijk niet-uniform rooster genereert. In het uiteindelijke rooster:  $h_{\min} = 0.0002125$  en  $h_{\max} = 1.05237$ , dus een verhouding  $h_{\max}/h_{\min} = 4950$ .

#### Diskussie

Wanneer we eenvoud van gebruik als een der belangrijke criteria zien, dan blijken deze routines van Pereyra zeer duidelijk gunstig uit te steken boven alternatieve aanpakken als schieten. De routines lossen "automatisch" het gestelde probleem op en bovendien doen ze dat efficiënt.

#### Literatuur

- LENTINI, M. & V. PEREYRA [1974], *A variable order finite difference method for nonlinear multipoint boundary value problems*, Math. Comp. 28, 981-1003.
- KELLER, H.B. [1974], *Accurate difference methods for nonlinear two-point boundary value problems*, SIAM J. Numer. Anal. 11, 305-320.
- PEREYRA, V. [1968], *Iterated deferred corrections for nonlinear boundary value problems*, Numer. Math. 11, 111-125.
- PEREYRA, V. & E.G. SEWELL [1975], *Mesh selection for discrete solution of boundary problems in ordinary differential equations*, Numer. Math. 23, 261-268.



Bijlagen

PROBLEEM 3.1.4.1  
M.B.V. PEIDE (NUMAL)

```

"BEGIN" "INTEGER" M, NN, NOBS, NBP;
"REAL" PAR [1 : 1], RES [1 : 1], JTJINV [1 : 1, 1 : 1], IN [0 : 6],
    OUT [1 : 8]; "INTEGER" "ARRAY" BP [0 : 1];
"REAL" TIME, FA, G, D;

"PROCEDURE" PEIDE (N, M, NO, NB, P, R, BP, J, I, O, D, JDY, JDP, CY,
    DA, MO); "CODE" 34444;
"PROCEDURE" COMMUNICATION (P, F, M, N, NO, NP, PA, R, BP, J, I, O, W,
    HI); "CODE" 34445;

"BOOLEAN" "PROCEDURE" JAC DFDP (PAR, Y, X, FP);
"REAL" X; "ARRAY" PAR, Y, FP;
"BEGIN" FP [1, 1] := FP [2, 1] := FP [3, 1] := 0;
    JAC DFDP := "TRUE"
"END" JAC DFDP;

"PROCEDURE" DATA (NOBS, TOBS, OBS, COBS);
"VALUE" NOBS; "INTEGER" NOBS; "ARRAY" TOBS, OBS, COBS;
"BEGIN"
    TOBS [0] := 0; TOBS [1] := 1500; COBS [1] := 1; OBS [1] := 0
"END" DATA;

"PROCEDURE" CALL YSTART (PAR, Y, YMAX);
"ARRAY" PAR, Y, YMAX;
"BEGIN" Y [1] := 0; Y [2] := 150; Y [3] := PAR [1];
    YMAX [1] := 1000; YMAX [2] := 150; YMAX [3] := 1;
    Y [21] := 1
"END" CALL YSTART;

"BOOLEAN" "PROCEDURE" DERIV (PAR, Y, X, DF);
"REAL" X; "ARRAY" PAR, Y, DF;
"BEGIN" "REAL" C, S, V;
    C := COS (Y [3]);
    S := SIN (Y [3]);
    V := Y [2];
    DF [1] := S / C;
    DF [2] := -G * S / (C * V) - D * V / C;
    DF [3] := -G / (V * V);
    DERIV := C > 0
"END" DERIV;

"BOOLEAN" "PROCEDURE" JAC DFDY (PAR, Y, X, FY);
"REAL" X; "ARRAY" PAR, Y, FY;
"BEGIN" "INTEGER" I, J; "REAL" C, S, V;
    C := COS (Y [3]);
    S := SIN (Y [3]);
    V := Y [2];
    "FOR" I := 1 "STEP" 1 "UNTIL" 3 "DO"
    "FOR" J := 1 "STEP" 1 "UNTIL" 3 "DO" FY [I, J] := 0;

```

```

    FY [1, 3] := 1 / (C * C);
    FY [2, 2] := (G * S / (V * V) - D) / C;
    FY [2, 3] := (-G / V - D * V * S) / (C * C);
    FY [3, 2] := 2 * G / (V * V * V);
    JAC DFDY := C > 0
"END" JAC DFDY;

"PROCEDURE" MONITOR (POST, NCOL, NROW, PAR, RES, WEIGHT, NIS);
"VALUE" POST, NCOL, NROW, WEIGHT, NIS;
"INTEGER" POST, NCOL, NROW, WEIGHT, NIS; "ARRAY" PAR, RES;
OUTPUT (61, "("/SZD, SB, N/)", POST, PAR [1]);

G := 9.80665; D := 0.00007;
PAR [1] := 1.15; NBP := 0;
IN [3] := "-6; IN [4] := "-6; IN [5] := S0; IN [6] := "-10;
IN [1] := "-4; IN [2] := "-6; FA := 0; IN [0] := "-14;
NN := 3; M := NOBS := 1; BP [0] := BP [1] := 0;
TIME := CLOCK;
COMMUNICATION (1, FA, NN, M, NOBS, NBP, PAR, RES, BP, JTJINV, IN,
               OUT, 0, 0);

PEIDE(NN, 1, 1, NBP, PAR, RES, BP, JTJINV, IN, OUT, DERIV, JAC DFDY,
      JAC DFDY, CALL YSTART, DATA, MONITOR);

TIME := CLOCK - TIME;
COMMUNICATION (2, FA, NN, M, NOBS, NBP, PAR, RES, BP, JTJINV, IN,
               OUT, 0, 0);
OUTPUT (61, "("/SB "("THE CALCULATION IN PEIDE CONSUMED")",
      BZZD.DBB, "("SECONDS")",*)", TIME)
"END"

```

STARTING VALUES OF THE PARAMETERS  
 $+1.150000 \times 10^1$

NUMBER OF EQUATIONS : 3  
 NUMBER OF OBSERVATIONS: 1

MACHINE PRECISION	$10^{-13}$
RELATIVE LOCAL ERROR BOUND FOR INTEGRATION	$10^{-5}$
RELATIVE TOLERANCE FOR RESIDUE	$10^{-5}$
ABSOLUTE TOLERANCE FOR RESIDUE	$10^{-5}$
MAXIMUM NUMBER OF INTEGRATIONS TO PERFORM	50
RELATIVE STARTING VALUE OF LAMBDA	$10^{-9}$
RELATIVE MINIMAL STEPLENGTH	$10^{-3}$

THERE ARE NO BREAK-POINTS

THE ALPHA-POINT OF THE F-DISTRIBUTION : 0.00

1	$+1.1500000000000 \times 10^0$
2	$+1.1500000000000 \times 10^0$
1	$+1.1500000000000 \times 10^0$
1	$+1.1548127086472 \times 10^0$
1	$+1.1549991219896 \times 10^0$
1	$+1.1550107739228 \times 10^0$
1	$+1.1550115120783 \times 10^0$
1	$+1.1550115588797 \times 10^0$
1	$+1.1550115618472 \times 10^0$
1	$+1.1550115620353 \times 10^0$

NORMAL TERMINATION OF THE PROCESS

LAST INTEGRATION WAS PERFORMED WITHOUT BREAK-POINTS

EUCL. NORM OF THE LAST RESIDUAL VECTOR ;.8609400" =7  
EUCL. NORM OF THE FIRST RESIDUAL VECTOR;.3300138" +2  
NUMBER OF INTEGRATIONS PERFORMED ; 8  
LAST IMPROVEMENT OF THE EUCLIDEAN NORM ;.1275210" =5  
CONDITON NUMBER OF J1\*J ;.1000000" +1  
LOCAL ERROR BOUND WAS EXCEEDED (MAXIM,); 0

THE LAST RESIDUAL VECTOR

I RES(I)  
1 +.8609" =7

THE CALCULATION IN PEIDE CONSUMED 43.77 SECONDS

C PROBLEEM 3.1.4.1  
 C M.B.V. D02ADF(NAG)  
 C

```

PROGRAM MASTER (OUTPUT, TAPE 6 = OUTPUT)
EXTERNAL DERIV, OUT
DIMENSION U (3, 2), V (3, 2), E (3), Y (6, 3), G (3, 3), INT (3),
1 WSP (3, 5), WS (3), YO (3), EE (3), AA (3), BB (3), CC (3), DD (3)
U (1, 1) = 0.0
V (1, 1) = 0.0
U (1, 2) = 0.0
V (1, 2) = 0.0
U (2, 1) = 150.0
V (2, 1) = 0.0
U (2, 2) = 135.0
V (2, 2) = 1.0
U (3, 1) = 1.15
V (3, 1) = 1.0
U (3, 2) = -1.2
V (3, 2) = 1.0
R = 750
X = 0.0
X1 = 1500.0
E (1) = 1E-2
E (2) = 1E-2
E (3) = 5E-5
H = 0.01
N = 3
M1 = 6
M = 1
WRITE (6, 98)
98  FORMAT (28H1  CURRENT PARAMETER VALUES,
1 4X, 23HERRORS IN Y[I] AT X = R, 18X, 17HSUM (ERRORS ** 2))
CALL D02ADF (U, V, E, Y, H, X, X1, R, N, M1, DERIV, OUT, G,
1 INT, WSP, WS, YO, EE, AA, BB, CC, DD, M)
IF (M .GT. 0) GOTO 3
WRITE (6, 99)
99  FORMAT (16H0 FINAL SOLUTION)
DO 2 I = 1, 6
WRITE (6, 100) I, (Y (I, J), J = 1, 3)
100 FORMAT (1H, I2, 3F10.4)
2  CONTINUE
3  WRITE (6, 101) M
101 FORMAT (13H ERROR NUMBER, I3)
10  CONTINUE
STOP
END

```

```
SUBROUTINE DERIV (G, Z, X)
DIMENSION G (3), Z (3)
GG = 9.80665
D = 0.00007
G (1) = SIN (Z (3)) / COS (Z (3))
G (2) = -GG * G (1) / Z (2) + D * Z (2) / COS (Z (3))
G (3) = -GG / (Z (2) * Z (2))
RETURN
END
```

```
SUBROUTINE OUT (A, B, F, R, N)
DIMENSION A (N, 2), B (N, 2), F (N), W (3)
J = 1
DO 3 K = 1, 2
DO 2 I = 1, N
IF (B (I, K) .EQ. 0.0) GOTO 2
W (J) = A (I, K)
J = J + 1
2 CONTINUE
3 CONTINUE
WRITE (6, 100) (W (J), J = 1, 3), (F (I), I = 1, 3), R
100 FORMAT (1H0, 3F9.4, 3E13.4, E15.4)
RETURN
END
```

CURRENT PARAMETER VALUES	ERRORS IN Y(I) AT X = R	SUM (ERRORS ** 2)
1.1500 135.0000 -1.2000	-.2927E+02    .2037E+01    .1334E+00	.8609E+03
1.1549 137.4487 -1.1905	-.5354E+00    .6099E+01    .2766E+02	.2904E+00
1.1551 137.5017 -1.1902	-.9342E+01    .1467E+01    .3008E+03	.8942E+02
1.1550 137.5021 -1.1902	.2198E+02    .3529E+03    .3118E+04	.4957E+05

FINAL SOLUTION

1	0.0000	150.0000	1.1550
2	555.2828	100.8934	.9588
3	848.4029	63.4423	.4800
4	863.1451	59.9158	.4053
5	586.3348	93.1578	.9592
6	0.0000	137.5021	-1.1902

ERROR NUMBER 0

PROBLEEM 3.1.4.2  
M.B.V. PEIDE (NUMAL)

```
"BEGIN" "INTEGER" I, J, M, NN, N, NOBS, NBP;
"ARRAY" PAR [1 : 1], RES [1 : 1], JTJINV [1 : 1, 1 : 1], IN [0 : 6],
      OUT [1 : 8]; "INTEGER" "ARRAY" BP [0 : 1], FAC [0 : 25];
"REAL" TIME, FA;

"PROCEDURE" PEIDE (N, M, NO, NB, P, R, BP, J, I, O, D, JDY, JDP, CY,
      DA, MO); "CODE" 34444;
"PROCEDURE" COMMUNICATION (P, F, M, N, NO, NP, PA, R, BP, J, I, O, W,
      NI); "CODE" 34445;

"REAL" "PROCEDURE" SUM (I, L, U, T); "VALUE" U;
"INTEGER" I, L, U; "REAL" T;
"BEGIN" "REAL" S; S:= 0;
      "FOR" I:= L "STEP" 1 "UNTIL" U "DO"
      S:= S + T; SUM:= S
"END" SUM;

"BOOLEAN" "PROCEDURE" JAC DFDP (PAR, Y, X, FP);
"REAL" X; "ARRAY" PAR, Y, FP;
"BEGIN" FP [1, 1]:= EXP (PAR [1]);
      JAC DFDP:= "TRUE"
"END" JAC DFDP;

"PROCEDURE" DATA (NOBS, TOBS, OBS, COBS);
"VALUE" NOBS; "INTEGER" NOBS; "ARRAY" TOBS, OBS, COBS;
"BEGIN"
      TOBS [0]:= 0; TOBS [1]:= 1; COBS [1]:= 1; OBS [1]:= 2
"END" DATA;

"PROCEDURE" CALL YSTART (PAR, Y, YMAX);
"ARRAY" PAR, Y, YMAX;
"BEGIN" Y [1]:= 0; YMAX [1]:= 4
"END" CALL YSTART;

"BOOLEAN" "PROCEDURE" DERIV (PAR, Y, X, DF);
"REAL" X; "ARRAY" PAR, Y, DF;
"BEGIN" "REAL" Y1, NLY, EMNLY;
      Y1:= Y [1]; NLY:= N * L (X) * Y1;
      EMNLY:= EXP (=NLY);
      DF [1]:= 1 + EXP (PAR [1])
      + LN (3) * (Y [1] * EMNLY * SUM (J, 0, N=2, NLY ** J / FAC [J])
      - EMNLY * SUM (J, 0, N=1, NLY ** J / FAC [J]) / L (X));
      DERIV:= "TRUE"
"END" DERIV;

"BOOLEAN" "PROCEDURE" JAC DFDY (PAR, Y, X, FY);
"REAL" X; "ARRAY" PAR, Y, FY;
"BEGIN" "REAL" Y1, NLY, EMNLY;
      Y1:= Y [1]; NLY:= N * L (X) * Y1;
      EMNLY:= EXP (=NLY);
      FY [1, 1]:= LN (3) * EMNLY * SUM (J, 0, N=1, NLY ** J / FAC [J]);
      JAC DFDY:= "TRUE"
"END" JAC DFDY;
```



```

"PROCEDURE" MONITOR (POST, NCOL, HROW, PAR, RES, WEIGHT, NIS);
"VALUE" POST, NCOL, HROW, WEIGHT, NIS;
"INTEGER" POST, NCOL, NROW, WEIGHT, NIS; "ARRAY" PAR, RES;
OUTPUT (61, ("/SZD,5B,N/"), POST, PAR [1]);

"REAL" "PROCEDURE" L (X); "VALUE" X; "REAL" X;
L:= 1 / (3 * X + 0.01);

FAC [0]:= M:= 1;
"FOR" I:=1 "STEP" 1 "UNTIL" 25 "DO" FAC [I]:= M:= M * I;

N:= 10;
PAR [1]:= 0; NBP:= 0;
IN [3]:= "-7; IN [4]:= "-7; IN [5]:= 50; IN [6]:= "-3;
IN [1]:= "-4; IN [2]:= "-7; FA:= 0; IN [0]:= "-14;
NN:= M:= NOBS:= 1; BP [0]:= BP [1]:= 0;
TIME:= CLOCK;
COMMUNICATION (1, FA, NN, M, NOBS, NBP, PAR, RES, BP, JTJINV, IN,
OUT, 0, 0);

PEIDE(1, 1, 1, NBP, PAR, RES, BP, JTJINV, IN, OUT, DERIV, JAC DFDY,
JAC DFDP, CALL YSTART, DATA, MONITOR);

TIME:= CLOCK - TIME;
COMMUNICATION (2, FA, NN, M, NOBS, NBP, PAR, RES, BP, JTJINV, IN,
OUT, 0, 0);
OUTPUT (61, ("/3/5B " ("THE CALCULATION IN PEIDE CONSUMED"),
BZZD.DDBB, ("SECONDS"),",,")", TIME)
"END"

```

STARTING VALUES OF THE PARAMETERS  
 +.0000000" +0

NUMBER OF EQUATIONS : 1  
 NUMBER OF OBSERVATIONS: 1

MACHINE PRECISION	;+.1"=13
RELATIVE LOCAL ERROR BOUND FOR INTEGRATION	;+.1" =6
RELATIVE TOLERANCE FOR RESIDUE	;+.10" =6
ABSOLUTE TOLERANCE FOR RESIDUE	;+.10" =6
MAXIMUM NUMBER OF INTEGRATIONS TO PERFORM	; 50
RELATIVE STARTING VALUE OF LAMBDA	;+.10" =2
RELATIVE MINIMAL STEPLENGTH	;+.10" =3

THERE ARE NO BREAK-POINTS

THE ALPHA-POINT OF THE F-DISTRIBUTION : 0.00

1	+0.0000000000000" +000
2	+0.0000000000000" +000
1	+0.0000000000000" +000
1	+4.0853574590190" =001
1	+4.0522950596640" =001
1	+4.0486250266894" =001
1	+4.0482209170912" =001
1	+4.0481764823811" =001
1	+4.0481715984346" =001
1	+4.0481710617162" =001

NORMAL TERMINATION OF THE PROCESS

LAST INTEGRATION WAS PERFORMED WITHOUT BREAK-POINTS

EUCL. NORM OF THE LAST RESIDUAL VECTOR :.1590742" =7  
EUCL. NORM OF THE FIRST RESIDUAL VECTOR :.8490653" +0  
NUMBER OF INTEGRATIONS PERFORMED ? 8  
LAST IMPROVEMENT OF THE EUCLIDEAN NORM :.1288565" =6  
CONDITON NUMBER OF J!\*J :.1000000" +1  
LOCAL ERROR BOUND WAS EXCEEDED (MAXIM.): 0

THE LAST RESIDUAL VECTOR

I	RES[I]
1	+.1591" =7

THE CALCULATION IN PEIDE CONSUMED 15.31 SECONDS

```

C   PROBLEEM 3.1.4.2
C   M.B.V. DO2AGF (NAG)
C

```

```

PROGRAM MASTER2 (OUTPUT, TAPE 6 = OUTPUT)
REAL MAT
DIMENSION PARAM (1), PARERR (1), ERROR (1), C (2, 1), MAT (1, 1),
1 COPY (1, 1), WSPACE (1, 9), G (1), G1 (1)
EXTERNAL AUX, PRSOL, RNAUX, BCAUX
PARAM(1)=0.0
N1=1
N=1
H=0.00001
ERROR(1)=1.0E-7
PARERR(1)=1.0E-7
M1=2
IFAIL=1
CALL DO2AGF (H,ERROR,PARERR,PARAM,C,N,N1,M1,AUX,BCAUX,
1 RNAUX,PRSOL,MAT,COPY,WSPACE,G,G1,IFAIL)
WRITE(6,9002)IFAIL
9002 FORMAT(1H0,7HIFAIL= ,I3/17HOFINAL PARAMETERS)
WRITE(6,9003)(PARAM(I),I=1,N1)
9003 FORMAT(1H ,3E16.8)
WRITE(6,9004)
9004 FORMAT(1H0,14HFINAL SOLUTION/26H X=VALUE AND COMPONENTS OF,
19H SOLUTION)
CALL RNAUX(X,X1,R,PARAM)
H = X1 = X
DO 1 I = 1, 2
DUM=X+FLOAT(I-1)*H
WRITE(6,9005)DUM,(C(I,J),J=1,N)
9005 FORMAT(1H ,F10.3,3E15.6)
1 CONTINUE
STOP
END

```

```

SUBROUTINE PRSOL(PARAM,RESID,N1,ERR)
DIMENSION PARAM(N1),ERR(N1)
WRITE(6,9000)(PARAM(1),ERR(1))
9000 FORMAT(12H PARAMETER 1,E13.7,/,12H RESIDU 1,E11.3,/)
RETURN
END

```

```

SUBROUTINE AUX(F,Y,X,PARAM)
REAL L, NLY
DIMENSION F (1), Y (1), PARAM (1)
N = 10
L = 1 / (3 * X + 0.01)
YY = Y (1)
NLY = N * L * YY
JFAC = 1
SOM = 0.0
NN = N - 2
DO 1 J = 1, NN
JFAC = JFAC * J
SOM = SOM + NLY ** J / JFAC
CONTINUE
SOM1 = SOM + 1.0
SOM2 = SOM1 + NLY ** (N - 1) / (JFAC * (N - 1))
F (1) = 1.0 + EXP (PARAM (1)) + ALOG (3.0) * EXP (-NLY) *
1 (YY * SOM1 - SOM2 / L)
RETURN
END

SUBROUTINE RNAUX(X,X1,R,PARAM)
DIMENSION PARAM(1)
X = 0.0
X1 = 1.0
R = 1.0
RETURN
END

SUBROUTINE BCAUX(G,G1,PARAM)
DIMENSION G(1),G1(1),PARAM(1)
G (1) = 0.0
G1 (1) = 2.0
RETURN
END

```

PARAMETER : 10.  
RESIDU : .849E+00

PARAMETER : .4779490E+00  
RESIDU : -.179E+00

PARAMETER : .3771742E+00  
RESIDU : .658E-01

PARAMETER : .4050347E+00  
RESIDU : -.528E-03

PARAMETER : .4048113E+00  
RESIDU : .838E-05

PARAMETER : .4048149E+00  
RESIDU : -.133E-06

IFAIL= 0

FINAL PARAMETERS  
.40481483E+00

FINAL SOLUTION  
X=VALUE AND COMPONENTS OF SOLUTION  
0.000 0.  
1.000 .200000E+01

PROBLEEM 3.1.4.3  
M.B.V. PEIDE (NUMAL)

```
"BEGIN" "INTEGER" M, NN, NOBS, NBP;
"ARRAY" PAR [1 : 1], RES [1 : 1], JTJINV [1 : 1, 1 : 1], IN [0 : 6],
      OUT [1 : 8]; "INTEGER" "ARRAY" BP [0 : 1];
"REAL" TIME, FA;

"PROCEDURE" PEIDE (N, M, NO, HB, P, R, BP, J, I, O, D, JDY, JDP, CY,
      DA, MO); "CODE" 34444;
"PROCEDURE" COMMUNICATION (P, F, M, N, NO, NP, PA, R, BP, J, I, O, W,
      NI); "CODE" 34445;

"BOOLEAN" "PROCEDURE" JAC DFDP (PAR, Y, X, FP);
"REAL" X; "ARRAY" PAR, Y, FP;
"BEGIN" FP [1, 1]:= FP [2, 1]:= 0;
      JAC DFDP:= "TRUE"
"END" JAC DFDP;

"PROCEDURE" DATA (NOBS, TOBS, OBS, COBS);
"VALUE" NOBS; "INTEGER" NOBS; "ARRAY" TOBS, OBS, COBS;
"BEGIN"
      TOBS [0]:= 0; TOBS [1]:= 1; COBS [1]:= 1; OBS [1]:= 0
"END" DATA;

"PROCEDURE" CALL YSTART (PAR, Y, YMAX);
"ARRAY" PAR, Y, YMAX;
"BEGIN" Y [1]:= 0; Y [2]:= PAR [1];
      YMAX [1]:= YMAX [2]:= 1;
      Y [14]:= 1
"END" CALL YSTART;

"BOOLEAN" "PROCEDURE" DERIV (PAR, Y, X, DF);
"REAL" X; "ARRAY" PAR, Y, DF;
"BEGIN"
      DF [1]:= Y [2];
      DF [2]:= EXP (Y [1]);
      DERIV:= "TRUE"
"END" DERIV;

"BOOLEAN" "PROCEDURE" JAC DFDY (PAR, Y, X, FY);
"REAL" X; "ARRAY" PAR, Y, FY;
"BEGIN"
      FY [1, 1]:= FY [2, 2]:= 0;
      FY [1, 2]:= 1; FY [2, 1]:= EXP (Y [1]);
      JAC DFDY:= "TRUE"
"END" JAC DFDY;

"PROCEDURE" MONITOR (POST, NCOL, NROW, PAR, RES, WEIGHT, NIS);
"VALUE" POST, NCOL, NROW, WEIGHT, NIS;
"INTEGER" POST, NCOL, NROW, WEIGHT, NIS; "ARRAY" PAR, RES;
OUTPUT (61, ("/5ZD, 5B, N/"), POST, PAR [1]);

PAR [1]:= 1; NBP:= 0;
IN [3]:= "-8; IN [4]:= "-8; IN [5]:= 50; IN [6]:= "-10;
IN [1]:= "-4; IN [2]:= "-8; FA:= 0; IN [0]:= "-14;
NN:= 2; M:= NOBS:= 1; BP [0]:= BP [1]:= 0;
TIME:= CLOCK;
```

```
COMMUNICATION (1, FA, NN, M, NOBS, NBP, PAR, RES, BP, JTJINV, IN,
              OUT, 0, 0);

PEIDE(NN, 1, 1, NBP, PAR, RES, BP, JTJINV, IN, OUT, DERIV, JAC DFDY,
      JAC DFDP, CALL YSTART, DATA, MONITOR);

TIME:= CLOCK - TIME;
COMMUNICATION (2, FA, NN, M, NOBS, NBP, PAR, RES, BP, JTJINV, IN,
              OUT, 0, 0);
OUTPUT (61, ("3/58 " ("THE CALCULATION IN PEIDE CONSUMED")",
            BZZD.DDBB, ("SECONDS")",*")", TIME)
"END"
```



STARTING VALUES OF THE PARAMETERS  
 $+1.0000000 \times 10^0$

NUMBER OF EQUATIONS : 2  
 NUMBER OF OBSERVATIONS: 1

MACHINE PRECISION	$\pm 1.0^{-13}$
RELATIVE LOCAL ERROR BOUND FOR INTEGRATION	$\pm 1.0^{-7}$
RELATIVE TOLERANCE FOR RESIDUE	$\pm 1.0^{-7}$
ABSOLUTE TOLERANCE FOR RESIDUE	$\pm 1.0^{-7}$
MAXIMUM NUMBER OF INTEGRATIONS TO PERFORM	: 50
RELATIVE STARTING VALUE OF LAMBDA	$\pm 1.0^{-9}$
RELATIVE MINIMAL STEPLENGTH	$\pm 1.0^{-3}$

THERE ARE NO BREAK-POINTS

THE ALPHA-POINT OF THE F-DISTRIBUTION : 0.00

1	$+1.0000000000000 \times 10^0$
2	$+1.0000000000000 \times 10^0$
1	$+1.0000000000000 \times 10^0$
1	$-2.5397516194423 \times 10^{-1}$
1	$-4.5315780618951 \times 10^{-1}$
1	$-4.6307915408640 \times 10^{-1}$
1	$-4.6360333442760 \times 10^{-1}$
1	$-4.6363109710557 \times 10^{-1}$
1	$-4.6363256767096 \times 10^{-1}$
1	$-4.6363264556597 \times 10^{-1}$

NORMAL TERMINATION OF THE PROCESS

LAST INTEGRATION WAS PERFORMED WITHOUT BREAK-POINTS

EUCL. NORM OF THE LAST RESIDUAL VECTOR ;.5051082" =8  
EUCL. NORM OF THE FIRST RESIDUAL VECTOR;.1841818" +1  
NUMBER OF INTEGRATIONS PERFORMED ; 8  
LAST IMPROVEMENT OF THE EUCLIDEAN NORM ;.9030730" =7  
CONDITON NUMBER OF J'\*J ;.1000000" +1  
LOCAL ERROR BOUND WAS EXCEEDED (MAXIM.); 0

THE LAST RESIDUAL VECTOR

I RES(I)  
1 +.5051" =8

THE CALCULATION IN PEIDE CONSUMED 18.96 SECONDS

PROBLEEM 3.1.4.3  
M.B.V. FEMLAG (NUMAL)

```

"BEGIN" "INTEGER" N, III; "REAL" XXX, YYY, C;
  "ARRAY" E [1 : 6];

  "PROCEDURE" FEMLAG (X, Y, N, R, F, ORDE, E); "CODE" 33301;

  "FOR" N:= 5, 10, 20 "DO"
  "BEGIN" "INTEGER" ORDE, I, IT; "REAL" NORM;
    "ARRAY" XX, YY, YN [0 : N];

    "REAL" "PROCEDURE" Y (X); "VALUE" X; "REAL" X;
    "IF" X = XXX "THEN" Y:= YYY "ELSE"
  "BEGIN" "INTEGER" I; "REAL" XXI, YYI;
    "IF" X < XX [III] "THEN" I:= 0 "ELSE" I:= III;
    "FOR" I:= I "WHILE" X > XX [I + 1] "DO" I:= I + 1;
    XXX:= X; III:= I; XXI:= XX [I]; YYI:= YY [I];
    YYY:= YYI + (YY [I + 1] - YYI) * (XXX - XXI) / (XX [I + 1] - XXI);
    Y:= YYY
  "END" Y;

    "REAL" "PROCEDURE" R (X); "VALUE" X; "REAL" X;
    R:= EXP (Y (X));

    "REAL" "PROCEDURE" F (X); "VALUE" X; "REAL" X;
  "BEGIN" "REAL" YX;
    YX:= Y (X); F:= EXP (YX) * (YX - 1)
  "END" F;

    "REAL" "PROCEDURE" NORM 2 (X, Y); "ARRAY" X, Y;
  "BEGIN" "INTEGER" I; "REAL" S;
    S:= 0;
    "FOR" I:= 0 "STEP" 1 "UNTIL" N "DO" S:= S + (X [I] - Y [I]) ** 2;
    NORM 2:= S
  "END" NORM 2;

  ORDE:= 2; III:= 0; XXX:= "+300"; C:= 0.66802784575;
  E [1]:= E [4]:= 1; E [2]:= E [3]:= E [5]:= E [6]:= 0;

  "FOR" I:= 0 "STEP" 1 "UNTIL" N "DO"
  "BEGIN" XX [I]:= I / N; YN [I]:= 0 "END";

  "FOR" IT:= 0, IT + 1 "WHILE" NORM > "=9, =4, =6 "DO"
  "BEGIN" "IF" IT < 0 "THEN" ORDE:= -IT;
  "FOR" I:= 0 "STEP" 1 "UNTIL" N "DO" YY [I]:= YN [I];
    FEMLAG (XX, YN, N, R, F, ORDE, E);
    NORM:= SQRT (NORM 2 (YY, YN));
    OUTPUT (61, "("("VERSCHIL MET VORIG RESULTAAT =")", 2ZD.9D,
      "(" ,ORDE V.D. METHODE = ")", ZD, /")", NORM, ORDE)
  "END";
  OUTPUT (61, "("//, 8B, "("X - WAARDE")", 9B, "("Y - WAARDE")",
    9B, "("EXACTE OPLOSSING")", 2/"");
  "FOR" I:= 0 "STEP" 1 "UNTIL" N "DO"
  OUTPUT (61, "("5B, 3 (=2ZD.9D, 5B), /")", XX [I], YY [I],
    2 * LN (C / COS (C * (XX [I] - 0.5))) + LN (2.0));
  OUTPUT (61, "("*)")
  "END"
"END"

```

VERSCHIL MET VORIG RESULTAAT = 0,184646567 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000836337 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000000016 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000000000 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000533397 ,ORDE V.D. METHODE = 4  
 VERSCHIL MET VORIG RESULTAAT = 0,000001004 ,ORDE V.D. METHODE = 6

X - WAARDE	Y - WAARDE	EXACTE OPLOSSING
0,000000000	0,000000000	=0,000000006
0,200000000	=0,073267595	=0,073268387
0,400000000	=0,109236582	=0,109237726
0,600000000	=0,109236582	=0,109237726
0,800000000	=0,073267595	=0,073268387
1,000000000	0,000000000	=0,000000006

VERSCHIL MET VORIG RESULTAAT = 0,261939722 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,001157893 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000000022 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000000000 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0,000190826 ,ORDE V.D. METHODE = 4  
 VERSCHIL MET VORIG RESULTAAT = 0,000000091 ,ORDE V.D. METHODE = 6

X - WAARDE	Y - WAARDE	EXACTE OPLOSSING
0,000000000	0,000000000	=0,000000006
0,100000000	=0,041435594	=0,041435629
0,200000000	=0,073268332	=0,073268387
0,300000000	=0,095799784	=0,095799853
0,400000000	=0,109237649	=0,109237726
0,500000000	=0,113703582	=0,113703662
0,600000000	=0,109237649	=0,109237726
0,700000000	=0,095799784	=0,095799853
0,800000000	=0,073268332	=0,073268387
0,900000000	=0,041435594	=0,041435629
1,000000000	0,000000000	=0,000000006

VERSCHIL MET VORIG RESULTAAT = 0.370668240 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0.001629765 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0.000000030 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0.000000000 ,ORDE V.D. METHODE = 2  
 VERSCHIL MET VORIG RESULTAAT = 0.000067653 ,ORDE V.D. METHODE = 4  
 VERSCHIL MET VORIG RESULTAAT = 0.000000008 ,ORDE V.D. METHODE = 6

X - WAARDE	Y - WAARDE	EXACTE OPLOSSING
0.000000000	0.000000000	=0.000000006
0.050000000	=0.021940976	=0.021940983
0.100000000	=0.041435621	=0.041435629
0.150000000	=0.058531193	=0.058531201
0.200000000	=0.073268379	=0.073268387
0.250000000	=0.085681701	=0.085681709
0.300000000	=0.095799844	=0.095799853
0.350000000	=0.103645927	=0.103645936
0.400000000	=0.109237717	=0.109237726
0.450000000	=0.112587791	=0.112587801
0.500000000	=0.113703652	=0.113703662
0.550000000	=0.112587791	=0.112587801
0.600000000	=0.109237717	=0.109237726
0.650000000	=0.103645927	=0.103645936
0.700000000	=0.095799844	=0.095799853
0.750000000	=0.085681701	=0.085681709
0.800000000	=0.073268379	=0.073268387
0.850000000	=0.058531193	=0.058531201
0.900000000	=0.041435621	=0.041435629
0.950000000	=0.021940976	=0.021940983
1.000000000	0.000000000	=0.000000006

C PROBLEM 3.1.4.3  
 C M.B.V. D02ADF (NAG)  
 C

```

PROGRAM MASTER (OUTPUT, TAPE 6 = OUTPUT)
EXTERNAL DERIV, OUT
DIMENSION U (2, 2), V (2, 2), E (2), Y (11, 2), G (2, 3), INT (2),
1 WSP (2, 5), WS (2), YO (2), EE (2), AA (2), BB (2), CC (2), DD (2)
U (1, 1) = 0.0
V (1, 1) = 0.0
U (1, 2) = 0.0
V (1, 2) = 0.0
U (2, 1) = 1.0
V (2, 1) = 1.0
U (2, 2) = -1.0
V (2, 2) = 1.0
C = 0.6680278475
R = 0.5
X = 0.0
X1 = 1.0
E (1) = 1E-7
E (2) = 1E-7
H = 0.000001
N = 2
M1 = 11
IFAIL = 1
WRITE (6, 98)
98 FORMAT (28H1 CURRENT PARAMETER VALUES,
1 4X, 23HERRORS IN Y[I] AT X = R, 6X, 17HSUM (ERRORS ** 2))
CALL D02ADF (U, V, E, Y, H, X, X1, R, N, M1, DERIV, OUT, G,
1 INT, WSP, WS, YO, EE, AA, BB, CC, DD, IFAIL)
IF (M .GT. 0) GOTO 3
WRITE (6, 99)
99 FORMAT (16H0 FINAL SOLUTION, 18X, 23H EXACT SOLUTION OF Y[1])
DO 2 I = 1, 11
X = (I - 1) / 10.0
YY = 2 * ALOG (C / COS (C * (X - 0.5))) + ALOG (2.0)
WRITE (6, 100) X, (Y (I, J), J = 1, 2), YY
100 FORMAT (1H, F4.1, 3F14.10)
2 CONTINUE
3 WRITE (6, 101) M
101 FORMAT (13H ERROR NUMBER, I3)
10 CONTINUE
STOP
END

```

```
SUBROUTINE DERIV (G, Z, X)
DIMENSION G (2), Z (2)
G (1) = Z (2)
G (2) = EXP (Z (1))
RETURN
END
```

```
SUBROUTINE OUT (A, B, F, R, N)
DIMENSION A (N, 2), B (N, 2), F (N), W (2)
J = 1
DO 3 K = 1, 2
DO 2 I = 1, N
IF (B (I, K) .EQ. 0.0) GOTO 2
W (J) = A (I, K)
J = J + 1
2 CONTINUE
3 CONTINUE
WRITE (6, 100) (W (J), J = 1, 2), (F (I), I = 1, 2), R
100 FORMAT (1H0, 2F14.10, 2E13.4, E15.4)
RETURN
END
```

CURRENT PARAMETER VALUES		ERRORS IN Y[I] AT X = R		SUM (ERRORS ** 2)
1.0000000000	-1.0000000000	.1549E=08	-.3372E+01	.1137E+02
-.4081784020	.4123032897	-.2144E=02	-.1193E+00	.1423E=01
-.4600197170	.4602415240	-.1152E=03	-.7816E=02	.6111E=04
-.4633930821	.4634068291	-.7141E=05	-.5192E=03	.2696E=06
-.4636166786	.4636175736	-.4649E=06	-.3452E=04	.1192E=08
-.4636325931	.4636325931	-.1616E=12	-.9497E=11	.9023E=22
-.4636325931	.4636325931	0.	-.1554E=14	.2416E=29
FINAL SOLUTION		EXACT SOLUTION OF Y[I]		
0.0	0.0000000000	-.4636325931	.0000000002	
.1	.0414356219	.3657558958	-.0414356231	
.2	.0732683795	-.2713999484	-.0732683816	
.3	.0957998449	.1795742504	-.0957998475	
.4	.1092377183	-.0893852455	-.1092377212	
.5	-.1137036534	.0000000009	-.1137036563	
.6	.1092377183	.0893852455	-.1092377212	
.7	.0957998449	.1795742504	-.0957998475	
.8	-.0732683795	.2713999484	-.0732683816	
.9	.0414356219	.3657558958	-.0414356231	
1.0	0.0000000000	.4636325931	.0000000002	
ERROR NUMBER 0				