DYNAMO perikelen

door:
H. Koppelaar, PSM-groep
St.Jacobsstraat 14
3511 BS Utrecht.

Inleiding

In de sociale wetenschappen wordt voor het simuleren van grote systemen wel gebruik gemaakt van de simulatietaal DYNAMO [1]. In dit artikel wordt het gebruik van DYNAMO bestudeerd aan de hand van een simulatie. We nemen hiertoe een eerste-orde systeem $x' = -x^2$, dat wil zeggen: de verandering in de grootheid x per tijdseenheid op een tijdstip t is gelijk aan het negatieve kwadraat van die grootheid op hetzelfde tijdstip. Het gedrag van x wordt bepaald door de analytische oplossing van dit eerste-orde systeem, namelijk $x(t) = 1/(1+t)$. Op het tijdstip $t = 0$ is x dus $x(0) = 1$, dit gebruiken we als startwaarde in de simulatie. De werkwijze om een simulatie van $x' = -x^2$ in DYNAMO te checken is dus door de exacte waarde $x(t) = 1/(1+1)$ successief te vergelijken met de gesimuleerde waarde $x^*(t)$ volgens de relatie $x' = -x^2$. De simulatie met behulp van DYNAMO vertoont een fout die te reduceren is met behulp van multistep methoden die juist vanwege de meerdere tijdstappen die daarvoor nodig zijn moeilijk in DYNAMO te programmeren zijn. Dit blijkt mede in dit artikel omdat het voor een multistep benodigde ALGOL-programma gepubliceerd wordt (zie de 4e paragraaf hierna). De benodigde wiskundige analyse voor een multistep is trouwens voor de meeste sociale wetenschappers te ingewikkeld, zoals mag blijken uit dit artikel. Daarom stap ik tenslotte over op de globale uitleg van een geavanceerd automatisch computerprogramma LEANS, in de hoop dat, ondanks de relatief moeilijke toegang tot dit pakket, ik de lezer overtuig dat nauwkeuriger analyse dan met DYNAMO mogelijk is, veel beter kan gebeuren met behulp van LEANS dan met de traditionele doe-het-zelf methode (= multisteps) uit de numerieke analyse.

De simulatie met DYNAMO

De eerste-orde vergelijking $x' = -x^2$ betekent:

$$\lim_{h \to 0} \frac{x(t+h) - x(t)}{h} = -x^2(t) \qquad \ldots(1)$$

De vergelijking (1) benaderen we voor voldoend kleine h met de Euler discretisatie

$$x(t+h) = x(t) - hx^2(t) \qquad \ldots(2)$$

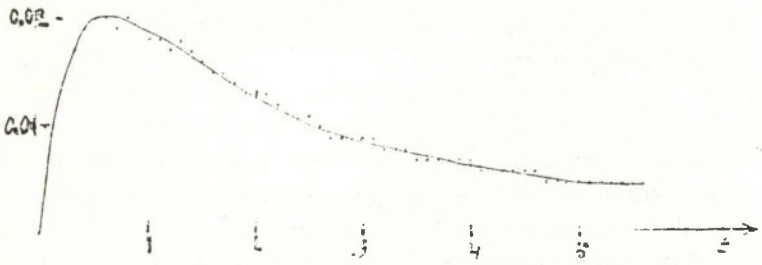
$$x(0) = 1 \qquad \ldots(3)$$

De DYNAMO statements worden dan:

```
1  DT  = .1;
2  T.K = T.J+DT;
3  T.O = 0;
4  X.K = X.J-DT*X.J**2;
5  X.O = 1;
6  Y.K = 1/(1+T.K);
```

Resultaten.

Als we de waarden van de Euler discretisatie (2) berekenen en vergelijken met de exacte uitkomsten, namelijk X.L = 1/(1+L) voor L = 0(0. 1)1 dus 100 stappen totdat TMAX = 10, dan krijgen we de volgende fout (figuur 1).



Sociale wetenschappers merken tegen mij wel op dat de nauwkeurigheid van een simulatie er niet toe doet, immers, de Sociale Wetenschappen verwachten slechts kwalitatief juiste antwoorden.

Echter, in het licht van D.E. Bailey's "The Siren Call to Distinguished Disaster" [2] hebben zij ongelijk. Daarom leg ik in dit artikel uit hoe de fout uit figuur 1 verbeterd kan worden.

Vanuit wiskundig standpunt kunnen de toepassingen van DYNAMO in twee categorieën verdeeld worden: (i) directe methoden, die een exact resultaat geven na een eindig aantal rekenstappen en (ii) benaderingswijzen, die slechts een benaderend resultaat geven na een eindig aantal rekenstappen. Indien de analytische oplossing van een wiskundig model (bijvoorbeeld een differentiaalvergelijking) niet bekend is, zal men altijd zijn toevlucht tot methoden van type (ii) moeten nemen. Terwijl (i) het nadeel heeft dat er afrondingsfouten in optreden, heeft (ii) het nadeel dat er èn afrondingsfouten èn representatiefouten (door de gehanteerde iteratiemethode) in optreden. De afrondingsfouten kunnen zeer ernstig accumuleren waartegen gewaarschuwd wordt door Bailey [2]. De fouten die ontstaan door de gehanteerde iteratie zijn heel wat moeilijker te analyseren; er zijn boeken aan gewijd [3]. In het vervolg wordt erop ingegaan hoe de discretisatie van de differentiaalvergelijking $x' = -x^2$ uit te voeren en daarmee te itereren zodat de fout in figuur 1 drastisch gereduceerd wordt.

Multistep methoden.

In de numerieke analyse is het - indien de analytische oplossing van een differentiaalvergelijking onbekend is - gebruikelijk om multistep discretisatie toe te passen. Omdat de Euler discretisatie slechts twee tijdstippen (steunpunten) t en t+h gebruikt en onnauwkeurige resultaten oplevert (zie figuur 1), ligt het voor de hand om meer steunpunten te gebruiken. Een veel gebruikte formule is er een met drie steunpunten, waarbij het middelste een zwaardere weging (namelijk factor 4) krijgt dan de andere twee. In algemene gedaante

is dit de multistep $x_{n+2} = x_n + \frac{h}{3}[\, f(t_{n+2}, x_{n+2}) + 4f(t_{n+1}, x_{n+1}) + f(t_n, x_n)]$, die we voor het gemak herschrijven tot:

$$x_n - x_{n-2} - (\tfrac{h}{3})(x'_n + 4x'_{n-1} + x'_{n+2}) = 0.$$

Taylor rond $t = t_{n-1}$ om de orde van deze multistep te bepalen:

$$x_n - x_{n-2} = 2h\, x'_{n-1} + 2\tfrac{h^3}{3!}\, x'''_{n-1} + 2\tfrac{h^5}{5!}\, x^{(5)}_{n-1} + O(h^6).$$

$$-(\tfrac{h}{3})(x'_n + 4x'_{n-1} + x'_{n-2}) = -2h\, x'_{n-1} - 2\tfrac{h^3}{3!}\, x'''_{n-1} - 2\tfrac{h^5}{3*4!}\, x^{(5)}_{n-1} + O(h^6).$$

$$x_n - x_{n-2} - (\tfrac{h}{3})(x'_n + 4x'_{n-1} + x'_{n-2}) = -\tfrac{4}{3*5!}\, h^5\, x^{(5)}_{n-1} + O(h^6).$$

Dus deze multistep is van de vierde orde en belooft veel nauwkeuriger resultaten.

We bekijken het stabiliteitsgebied. De multistep levert voor de differentiaalvergelijking $x' = \lambda x$ de basisoplossingen:

$$x_n = \sum_{i=1}^{p} c_i r_i^n \text{ met } r_{1,2} = \frac{\frac{2}{3} h\lambda \pm \sqrt{1 + \frac{1}{3} h^2\lambda^2}}{1 - \frac{1}{3} h\lambda}$$

Als $h \to 0$ is $|r_1| = |r_2| = 1$, de methode is dus niet sterk stabiel, maar nog wel stabiel. Volgens Van der Sluis [4] zullen fouten in de wortels zich vrijwel als oplossingen van de hom. differentiaalvergelijking gaan voortplanten als voldaan is aan:

$$\left|\frac{r_2}{r_1}\right| = \left|\frac{\frac{2}{3} h\lambda - \sqrt{1 + \frac{1}{3}(h\lambda)^2}}{\frac{2}{3} h\lambda + \sqrt{1 + \frac{1}{3}(h\lambda)^2}}\right| < 1.$$

Hierbij is voor $h\lambda > 0$: $\left|\frac{r_2}{r_1}\right| < 1$

en voor $h\lambda < 0$: $\frac{r_2}{r_1} > 1$

Dus deze methode is absoluut stabiel op intervallen $[0,\alpha]$, waarbij $\alpha > 0$, met andere woorden: het absolute stabiliteitsgebied van deze methode en het complexe linkerhalfvlak zijn disjuncte verzamelingen. De praktische consequentie hiervan is dat voor differentiaalvergelijkingen waarbij $\lambda < 0$, welke een exacte homogene oplossing hebben die uitsterft voor $t \to \infty$, aan de eis voor deze methode

$|\frac{r_2}{r_1}| = |\frac{s}{r}| < 1$ niet voldaan is.

Integendeel, $|\frac{r_2}{r_1}| = |\frac{s}{r}| > 1$, zodat de multistep een stijgende functie $x_n = Ar_1^n + Br_2^n$ oplevert, terwijl de oplossingen voor $x' = \lambda x$, met $\lambda < 0$ juist dalend zijn. Voor $\lambda < 0$ gaat de methode dus grove fouten maken.

De gekozen multistep heeft dus slechte eigenschappen voor ons probleem. We proberen nu een multistep die een predictor is, dat wil zeggen waarin het jongste tijdstip slechts eenmaal voorkomt. Met Taylorontwikkeling rond $t = t_{n-1}$ vinden we dat de volgende predictor van de orde 3 is.

$$x_n + 4x_{n-1} - 5x_{n-2} - 2h\,(2x'_{n-1} + x'_{n-2}) = 0$$

De karakteristieke polynoom van deze recursie is:

$$r^2 + 4(1-h\lambda)r - (5 + 2h\lambda) = 0$$

zodat $r_{1,2} = -2(1 - h\lambda) \pm \sqrt{9 - 6h\lambda + 4(h\lambda)^2}$

Het plusteken geldt voor $r_1 \approx 1 + h\lambda + O(h^2)$ voor kleine h.

Als $h \to 0$ dan $r_2 \to -5$, de predictor is dus niet stabiel. Dit loopt uit de hand. Daarom voeren we nu met deze predictor en de daarvoor gegeven multistep een corrector iteratie uit, in de hoop dat dit een goede greep is:

predictor: $x_{n+1}^{(0)} = -4x_n + 5x_{n-1} + (2h)(2x_n' + x_{n-1}')$

$[x_{n+1}^{(0)}]' = f(t_{n+1}, x_{n+1}^{(0)})$

corrector: $x_{n+1}^{(1)} = x_{n-1} + (\frac{h}{3})\{[x_{n+1}^{(0)}]' + 4x_n' + x_{n-1}'\}$

Taylorontwikkeling rond $t = t_{n-1}$ toont aan dat deze methode van de 4e orde is:

$$x_n - x_{n-2} - 2h\, x_{n-2}' - \frac{2}{3} h^2(2x_{n-1}'' + x_{n-2}'') = 0,$$

$$x_n - x_{n-2} = 2h\, x_{n-1}' + 2\,\frac{h^3}{3!}\, x_{n-1}''' + 2\,\frac{h^5}{5!}\, x_{n-1}^{(5)} + O(h^6),$$

$$-2h\, x_{n-2}' - \frac{4}{3} h^2\, x_{n-1}'' - \frac{2}{3} h^2\, x_{n-2}'' = -2h\, x_{n-1}' - \frac{1}{3} h^3\, x_{n-1}''' + \frac{1}{36} h^5\, x_{n-1}^{(5)} + O(h^6),$$

dus $x_n - x_{n-2} - 2h\, x_{n-2}' - \frac{2}{3} h^2(2x_{n-1}'' + x_{n-2}'') = \frac{4}{90} h^5\, x_{n-1}^{(5)} + O(h^6).$

De predictor-correctormethode luidt in zijn geheel:

$$x_n - x_{n-2} - 2h\, x_{n-2}' - \frac{2}{3} h^2(2x_{n-1}'' + x_{n-2}'') = 0.$$

De karakteristieke vergelijking voor deze recursie is:

$$r^2 - \tfrac{4}{3}(h\lambda)^2\, r - (1 + 2h\lambda + \tfrac{2}{3}(h\lambda)^2) = 0,$$

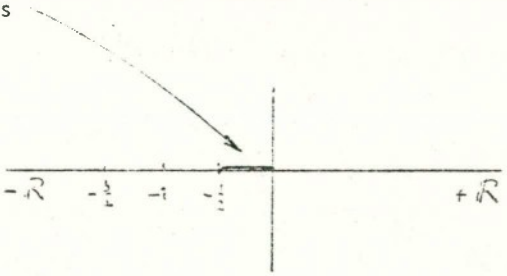
zodat $r_{1,2} = \frac{2}{3}(h\lambda)^2 \pm \frac{1}{2}\ \sqrt{3 + 8h\lambda + (1 + \frac{4}{3}(h\lambda)^2)^2}$

Het plusteken geldt voor $r_1 \approx 1 + \lambda\, 0(h)$ voor kleine h. Als $h \to 0$ dan geldt voor beide $|r_1| = |r_2| = 1$, de methode is dus niet <u>sterk</u> stabiel, maar nog wel stabiel.
Analoog aan de beschouwingen voor de eerste multistep bekijken we nu:

$$\left|\frac{r_2}{r_1}\right| = \left| \frac{\frac{4}{3}(h\lambda)^2 - \sqrt{3} + 8h\lambda + (1 + \frac{4}{3}(h\lambda)^2)^2}{\frac{4}{3}(h\lambda)^2 + \sqrt{3} + 8h\lambda + (1 + \frac{4}{3}(h\lambda)^2)^2} \right| < 1$$

Hierbij is voor $h\lambda > -\frac{1}{2}$: $|\frac{r_2}{r_1}| < 1$ en $\lambda \varepsilon R$

De doorsnede van het absolute stabiliteitsgebied met de negatieve reële as is



Vergeleken met de eerst gekozen multistep hebben we nu heel wat gewonnen. De predictor-corrector methode is stabiel en maakt geen fouten voor $\lambda < 0$ in $x' = \lambda x$. We hanteren deze methode in een simulatie.

De simulatie zonder DYNAMO

Om een multistep te kunnen hanteren in een berekening (simulatie) moet de enkelvoudige startwaarde $x(0) = 1$ uitgebreid worden tot meerdere startwaarden. Dit gebeurt met behulp van een Runge-Kutta schema [3]. De theorie hiervoor blijft achterwege. Ik volsta met het startalgorithme en de predictor-corrector methode daaronder gevoegd. Dit alles is geschreven in ALGOL60, omdat DYNAMO het mij hier te moeilijk maakt vanwege de nogal zware syntaxrestricties.

```
begin integer i,j,n; real f,h; array k [1:4];

comment hierna komt de stapgrootte h en het aantal stappen n;
        h: = 0.2; n:= 100/h;

begin array y[0:n ]; y [ 0] : = 1;

      k [1] : =-h x(y [0] ↑ 2);

      k [2] : =-h x(y [0] + 0.4 x k [1]) ↑ 2;

      k [3] : =-h x(y [0] + 0.29697760 x k [1] + 0.15875966 x k [2] ) ↑ 2;

      k [4] : =-h x(y [0] + 0.21810038 x k [1] - 3.05096470 x k [2] +
                                              + 3.83286432 x k [3]) ↑  2;

      y [1] : = y [0] + 0.17476028 x k [1] - 0.55148053 x k [2] +
                + 1.20553547 x k [3] + 0.17118478 x k [4] ;

comment  nu komt de predictor-corrector;

for j: = 0 step 1 until n-2 do

begin predict:

f: =-4 x y [j+1] + 5 x y [j] + 2 x h x (-2 x y [j+1] ↑ 2 - y [j] ↑ 2);

correct:

y [j+2] : = y [ j] + h/3 x (-f ↑ 2 - 4 x y[j+1] ↑ 2 - y [ j] ↑ 2);

end end end
```

Resultaten van de tweede simulatie

De fout in de oplossing verkregen met de predictor-corrector methode

en Runge-Kutta start neemt toe met toenemende stapgrootte h en neemt af met toenemende t. Dit is te zien omdat de predictor-corrector methode een afbreekfout levert:

$$-(4/90)\ h^5\ x^{*(5)} = (16/3)\ h^5(1+t)^{-6}$$

De afbreekfout sterft dus uit en is maximaal voor $t = nh < 1$. Voor deze identiteit is overigens gebruik gemaakt van:

$$\frac{d^n x}{dt^n} = (-1)^n\ n!\ x^{n+1} \text{ met } x(t) = (1+t)^{-1}$$

De foutschatting kan nauwkeuriger, zie Van der Sluis [4]. De resultaten zijn echter zoveel beter dan in de eerste simulatie met behulp van de Euler discretisatie in DYNAMO dat ik van nauwkeuriger schattingen afzie. Een plot van de berekende oplossing is te vinden in figuur 2.

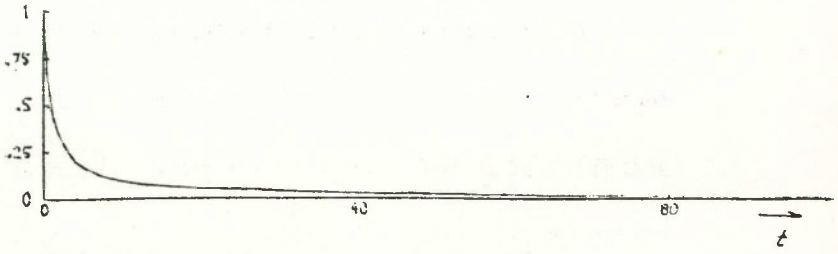


fig. 2 de oplossing van $x' = -x^2$

Automatisering van de verbeterde aanpak.

Mijn kritiek op de maximaal haalbare nauwkeurigheid in DYNAMO verleidde mij tot het formuleren van een multistep aanpak als vervanging van deze programmeertaal. Voor de met opzet zo simpel gekozen differentiaalvergelijking was er toch nog vrij veel analyse nodig om tot een goede methode te komen. Wat moet dat worden als de modellen ingewikkelder worden en zelfs niet lineair zijn? Het antwoord hierop is niet geheel afdoende, maar voor een grote klasse van modellen is er een automatisch simulatiepakket dat veel beter is dan DYNAMO, namelijk: LEANS. Voor sociale wetenschappers is dit pakket enigszins lastig te gebruiken omdat de invoer van gegevens over het te simuleren model gedaan moet worden in termen van een 'blokschema'. LEANS (= Lehigh-University Analog Simulator) behoort met MIDAS en 1130-CSMP tot een groep simulatie-programma's waarbij een gemakkelijke beschrijving van het te simuleren systeem in blokschema mogelijk is. Blokschema's komen uit de electronica en zijn oorspronkelijk bedoeld voor analoge schakelingen. Het hoofdidee achter LEANS-versie G: LEAG is dan ook om met een digitale computer een analoog model door te rekenen, mits dat model in blokvorm aangeboden wordt. Digitale simulatie van continue of intermitterend werkende systemen heeft verschillende voordelen boven analoge simulatie (op een analoge simulator): de resultaten liggen reproduceerbaar vast, de variabelen hebben een vrijwel onbeperkt groottebereik (geen schaalproblemen of overbelastingen) en de resultaten kunnen door dezelfde computer statistisch geëvalueerd worden (bijvoorbeeld via SPSS). Een digitale computer is veelal gemakkelijker beschikbaar dan een analoge. Een bezwaar is daarentegen dat nauwkeurigheid alleen bereikt kan worden ten koste van snelheid: snel repeterend rekenen is niet mogelijk. LEANS is efficiënt in FORTRAN-IV geschreven in biedt de mogelijkheid middelgrote modellen (tot 300 blokken en 75 integratoren) op een grote digitale computer

te simuleren. De ontwerpers van LEANS-versie A zijn Morris en Schiesser in 1965. LEANS-G is ten opzichte van de vorige versies uitgebreid met een zeer efficiënte automatische methode voor lineaire systemen. Deze methode zullen we hier beschrijven en toepassen op het iets ingewikkelder model $y'' = -y$. Het zal duidelijk worden dat er dan heel wat gewonnen is aan programmeereenvoud ten opzichte van de gepropageerde multistep-methoden; daartegenover staat dat de wiskunde achter deze LEANS werkwijze een tikje moeilijker is.

Van model tot blokschema.

Om het model $y'' = -y$ te kunnen simuleren met LEANS, moeten we letten op de beperkingen die LEANS heeft: de formulering moet in blokvorm, met behulp van de zogenaamde 'functieblokken' en 'integratoren'. Iedere integrator kan slechts een differentiaalvergelijking van de eerste orde aan, dus de tweede-orde differentiaalvergelijking $y'' = -y$ moet herschreven worden in twee eerste-orde differentiaalvergelijkingen als volgt:

Stel $x_0 = x_0(t) = y(t)$ en

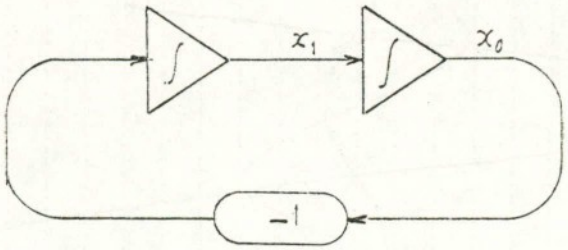
$$x_1 = x_1(t) = \frac{d}{dt} y(t)$$

Hiermee vinden we als nieuwe formulering het model:

$$\frac{d}{dt} x(t) = Ax(t)$$

waarin $x(t)$ een vector is $(x_0(t), x_1(t)^T$ en $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$

De blokvorm wordt nu:



Uit dit blokschema is af te lezen dat x_1 eenmaal geïntegreerd wordt om x_0 te verkrijgen. Dit is hetzelfde als $x_1 = \frac{d}{dt} x_0$ (op een constante na), verder is eruit af te leiden dat x_0 maal -1 en eenmaal geïntegreerd, gelijk moet zijn aan x_1, dus $\frac{d}{dt} x_1 = -x_0$.

Het programma in LEANS voor dit model luidt nu in hoofdlijnen als volgt:

ELEMENT TYPE	BLOCK NO.	INPUT
Integrator	1	2
Integrator	2	3
Gain	3	1

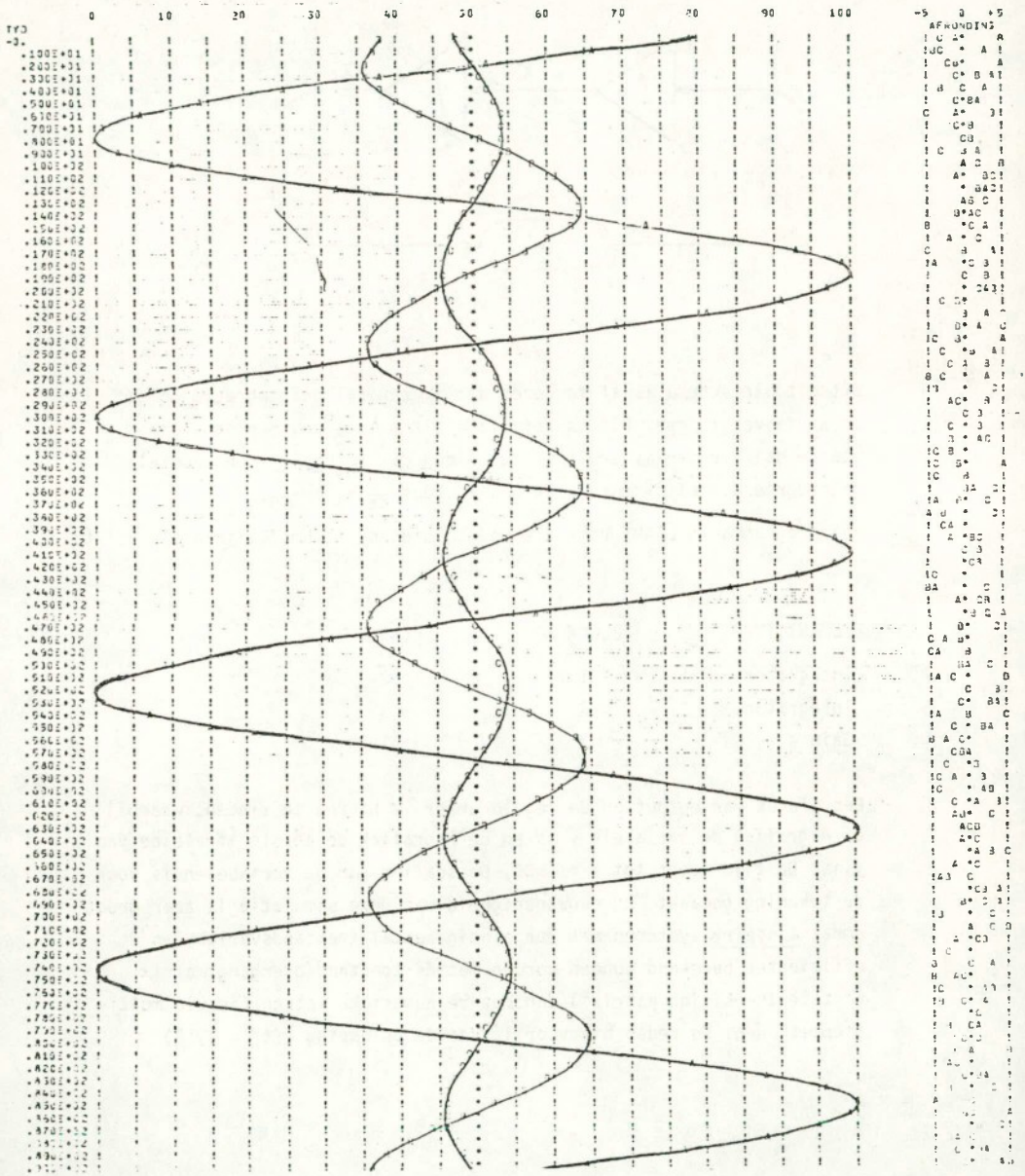
Een afdruk van output op de regeldrukker is hierna te vinden, waarbij de A-grafiek de variabele y is en de B-grafiek de eerste afgeleide van y is. De tijd loopt tot t = 900. De scaling van de variabelen is voor de tekening gemaakt. De nauwkeurigheid van deze simulatie is zeer groot omdat lineaire systemen met een eindig aantal toestandsvariabelen efficiënter berekend kunnen worden met de toestandsovergangsmatrix ('state transition matrix') dan met de numerieke integratie via multistepmethoden. De reden hiervoor is dat de oplossing $y(t) = \Phi y(0)$

DE GRAFIEKEN STAAN IN EEN ONDERLING JUIST VERBAND.

DE WAARDE VAN PRINTPOSITIE N LIGT TUSSEN MINIMUM+(N-0.5).SCHAALDEEL EN MINIMUM+(N+0.5).SCHAALDEEL

FUNC	SYMB	INTERVAL	MAXIMUM	MINIMUM	SCHAALDEEL
1	A	.693E+01	.346E+01	-.346E+01	.693E-01
2	B	.196E+01	.970E+00	-.980E+00	.693E-01
3	C	.554E+00	.277E+00	-.277E+00	.693E-01

POS. X-AS



van een lineair systeem te berekenen is via Taylor-ontwikkeling van $\Phi = \exp(At)$, waarbij A de matrix

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

is. Deze methode is allang bekend in de wiskunde, maar is in het pakket LEAG automatisch ingebouwd. Het programma zoekt zelf in de opgegeven probleemrepresentatie of deze methode toepasbaar is, zie Goldberg. Zoals gezegd is de wiskunde voor LEANS wat ingewikkelder, maar het gebruik is eenvoudig.

Literatuur

1. H. Koppelaar, W.M. de Pijper, DYNAMO, Accuflodder nr. 26, Utrecht, 1977.

2. D.E. Bailey,'The Siren Call to Distinguished Disaster', in: Sociaal Wetenschappelijke Informatica, D.E. Bailey (author), Amsterdam, 1978.

3. H.J. Stetter, Analysis of Discretization Methods for Ordinary Differential Equations, Springer Tracts in Nat. Phil., Volume 23, Berlin, 1973.

4. A. van der Sluis, Dictaat Numerieke Analyse, Utrecht, 1972.

5. R. van Houten, 'LEANS', ACCUflodder nr. 1, ACCU, Budapestlaan 6, Utrecht, 1975.

6. J.L. Goldberg, A.J. Schwartz, Systems of Ordinary Differential Equotions, Harper & Row, New York, 1972.