

DR. P. J. VAN DER HOUWEN

DE NUMERICUS
EN DE COMPUTER

Rede

*uitgesproken bij de aanvaarding van het
ambt van bijzonder hoogleraar in de
numerieke wiskunde en informatica
aan de Universiteit van Amsterdam
vanwege de Stichting voor Hoger Onderwijs
in de Toegepaste Wiskunde
op maandag 25 oktober 1976*

DE NUMERICUS EN DE COMPUTER

DE NUMERICUS EN DE COMPUTER

*Rede
uitgesproken bij de aanvaarding van het
ambt van bijzonder hoogleraar in de
numerieke wiskunde en informatica
aan de Universiteit van Amsterdam
vanwege de Stichting voor Hoger Onderwijs
in de Toegepaste Wiskunde
op maandag 25 oktober 1976*

door

Dr. P. J. van der Houwen

1976
Van Gorcum, Assen/Amsterdam

© 1976 Van Gorcum & Comp. B.V., Postbus 43, ASSEN

Uit deze uitgave mag niets worden gereproduceerd d.m.v. boekdruk, foto-offset, fotokopie, mikrofilm of welk ander medium dan ook, zonder schriftelijke toestemming van de uitgevers.

I S B N 90 232 1468 8

Layout en druk: Van Gorcum, Assen

*Dames en Heren leden van de Universiteitsraad
en van het College van Bestuur,
Mijne Heren curatoren en bestuursleden van de
Stichting voor Hoger Onderwijs in de Toegepaste
Wiskunde,
Dames en Heren leden van de Wetenschappelijke
Staf,
Dames en Heren leden van de Technische en Ad-
ministratieve Staf,
Dames en Heren studenten,
En voorts Gij allen die door Uw aanwezigheid
blijk geeft van Uw belangstelling,*

Zeer geachte toehoorders,

U zult zich allen uit Uw schooltijd nog wel de formule herinneren waarmee de wortels van een vierkantsvergelijking berekend kunnen worden. Behalve de vier rekenkundige operaties optellen, aftrekken, vermenigvuldigen en delen, treffen we in deze formule de operatie worteltrekken aan en velen van U zullen zich ook nog wel de rekenkunstjes herinneren om de wortel uit een getal te trekken, waarmee de operatie worteltrekken teruggebracht wordt tot een reeks van de zojuist genoemde rekenkundige operaties. Het aldus verkregen rekenschema, uitsluitend bestaande uit optellingen, aftrekkingen, vermenigvuldigingen en delingen, is een voorbeeld van een numerieke algoritme.

In de numerieke wiskunde houdt men zich bezig met het ontwerpen en analyseren van numerieke

algoritmen voor de oplossing van technisch-wetenschappelijke problemen. Heeft men voor het gegeven probleem een algoritme opgesteld, dan hoeven we niet meer na te denken en kunnen we de oplossing van het probleem vinden door eenvoudig de reeks operaties uit te voeren waarmee de algoritme gedefinieerd is. Dit is een zo automatisch proces dat we de uitvoering hiervan kunnen uitbesteden aan een rekenautomaat. Ik denk hierbij niet aan de zakrekenmachientjes die de laatste jaren in opmars zijn, maar aan de programmeerbare rekenmachines die lange reeksen operaties zonder tussenkomst van de mens kunnen uitvoeren. Uiteraard moet de rekenmachine wel „verteld” worden wat van hem verwacht wordt; daartoe vertaalt men de algoritme in een voor de rekenmachine begrijpelijke taal. Numerieke algoritmen geschreven in een vorm welke direct door een rekenmachine verwerkt kan worden, noemt men numerieke programmatuur.

Deze rede is gewijd aan de ontwikkeling van numerieke programmatuur en haar plaats in de numerieke wiskunde. Na een kort overzicht van feiten die significant zijn geweest voor haar ontstaan, zal ik een aantal aspecten van de vervaardiging van programmatuur bespreken. Dit zal ons vanzelf leiden tot de vraag hoe ver men gevorderd is met – wat men zou kunnen noemen – de programmering van de numerieke wiskunde. In het bijzonder zal ik op deze vraag ingaan waar het programma voor differentiaalvergelijkingen betreft, deels omdat dit gebied van de numerieke wiskunde mijn speciale interesse heeft, maar vooral ook omdat hier juist de laatste jaren een onstuimige ontwikkeling plaatsvindt.

1. Het ontstaan van numerieke programmatuur

De eerste publicaties in het in 1943 opgerichte tijdschrift *Mathematical Tables and other Aids to Computation*, welke gewijd waren aan tabellen voor speciale functies uit de mathematische fysica, noemt Rice¹ de numerieke programmatuur van voor het computertijdperk. Hij merkt er echter direct bij op dat er weinig gemeenschappelijke elementen zijn met wat men nú numerieke programmatuur noemt. De enige overeenkomst is, dat men niet zélf de reeks bewerkingen behoeft uit te voeren waarmee de algoritmen voor de berekening van deze speciale functies zijn gedefinieerd. Een van de eerste numerieke algoritmen die op een rekenmachine uitgevoerd kon worden, werd in 1946 door Alt² gepubliceerd. Deze algoritme betrof de vermenigvuldiging van matrices. Op een IBM-machine kon hiermee de vermenigvuldiging van twee matrices van de orde 9 – dit betekent ongeveer 1400 optellingen en vermenigvuldigingen – in 9 minuten uitgevoerd worden.

In de hierop volgende tien jaren werden steeds meer algoritmen geprogrammeerd. Omdat de programmering in machinecode plaatsvond, dus machine-afhankelijk was, moesten deze algoritmen echter voor elke nieuwe rekenmachine die op de markt kwam, opnieuw geïmplementeerd worden. Mede doordat juist in deze periode de rekenmachine een stormachtige ontwikkeling doormaakte, bleef de verzameling van geprogrammeerde algoritmen, tegenwoordig programmatheek genoemd, bescheiden van omvang.

Een mijlpaal in de ontwikkeling van grotere programmatheken is de introductie van gestandaardiseerde algoritmische programmeertalen geweest. Dit zijn machine-onafhankelijke talen waarin al-

goritmen zich in zeer leesbare vorm laten formuleren. Een eerste stap in de richting van deze hogere programmeertalen werd in 1957 gedaan met het invoeren van de eerste versie van de taal FORTRAN. Een jaar later werd de taal IAL, een afkorting van *International Algorithmic Language*, gedefinieerd, waaruit het latere ALGOL 60³ ontstaan is. Hierna zijn vele andere hogere programmeertalen ontworpen. Voor de numerieke wiskunde zijn naast FORTRAN en ALGOL nog van belang de talen PASCAL⁴, SIMULA⁵ en ALGOL 68⁶.

Eigenlijk is pas na de introductie van machine-onafhankelijke programmeertalen de numerieke programmatuur een belangrijke tak in de numerieke wiskunde geworden. Zo is er sinds 1960 aan de *Communications of the Association for Computing Machinery* een aparte bijlage⁷ toegevoegd voor de publicatie van in ALGOL, en sinds 1967 ook van in FORTRAN, geschreven algoritmen. Het aantal gepubliceerde algoritmen is onlangs de 500 gepasseerd. Deze omvangrijke verzameling van losse stukken programmatuur vormt echter niet bepaald een programmatheek die men de gebruiker zou willen aanbieden. De eerste serieuze poging om te komen tot een samenhangende programmatheek van numerieke algoritmen dateert uit 1960 toen het juist verschenen tijdschrift *Numerische Mathematik*, het *Handbook for Automatic Computation* aankondigde. Dit Handbook beoogde te komen tot een verzameling geteste algoritmen voor wiskundige berekeningen geschreven in een algemene programmeertaal. Dat drie van de zes redacteurs van het Handbook-project medeauteur zijn van het officiële ALGOL 60 rapport, zal er niet vreemd aan zijn dat gekozen is voor ALGOL 60 als programmeertaal. Het eerste

deel van het Handbook is gewijd aan de definitie en implementatie van ALGOL 60; het tweede deel, dat in 1971⁸ verscheen, bevat procedures op het gebied van de lineaire algebra. Het feit dat een tiental jaren verstreken zijn, voordat dit vrij beperkte procedurepakket gepubliceerd kon worden, is een aanwijzing dat de ontwikkeling van programmatheken tot de grotere projecten gerekend dient te worden; men meet dergelijke projecten in vele manjaren⁹.

De ontwikkeling van numerieke programmatuur als zelfstandige tak van wetenschap krijgt steeds meer erkenning. In 1974 richtte de *International Federation for Information Processing de Working Group 2.5 on Numerical Software* op met het doel de kwaliteit van numerieke berekeningen te verbeteren, door de ontwikkeling en beschikbaarstelling van hoog gekwalificeerde numerieke programmatheken te bevorderen. Deze groep telt leden uit de omgeving van belangrijke rekencentra uit de gehele wereld. Met de oprichting van deze werkgroep is een begin gemaakt met een internationale samenwerking op het gebied van de numerieke programmatuur.

2. De ontwikkeling van numerieke programmatheken

Na deze historische schets wil ik nader ingaan op de ontwikkeling van een numerieke programmatheek. Een programmatheek dient een "verlengstuk" te zijn van de programmeertaal waarin zij geschreven is, een soort van gereedschapskist waarmee wiskundige problemen opgelost kunnen worden. De stukken gereedschap worden procedures of routines genoemd. Uiteraard zal een pro-

grammatheek niet voor elk willekeurig probleem een aparte procedure kunnen bevatten, maar veel-
eer bestaan uit een verzameling procedures waar-
mee een reeks van probleemklassen opgelost
kunnen worden.

Zoals al opgemerkt is, moet men de ontwikkeling
van een goed voorziene programmatheek tot de
grotere projecten rekenen. In zo'n project kan
men een aantal fasen onderscheiden:

- 1e De selectie van numerieke berekeningsme-
thoden
- 2e De formulering van de berekeningsmethode
als een algoritme en de programmering in een pro-
grammeertaal
- 3e De documentatie van de procedures

De aspecten *selectie* en *programmering* wil ik wat
nader toelichten.

Bij de selectie van een numerieke berekeningsme-
thode heeft men in eerste instantie te maken met
stabiliteit, *betrouwbaarheid* en *efficiëntie*. Stabili-
teit is hier in de eerste plaats genoemd, omdat de
methode voor een rekenmachine met eindige pre-
cisie geïmplementeerd dient te worden en er dus
tegen bestand moet zijn dat rekenkundige opera-
ties niet exact uitgevoerd worden. Het is daarom
een eerste vereiste dat de gekozen methode zo sta-
biel is, dat ze min of meer ongevoelig is voor dit
soort verstoringen. Met het tweede criterium, be-
trouwbaarheid, wordt bedoeld dat de methode
voldoet aan de opgegeven specificaties. Een illus-
tratie hiervan vormt de in veel methoden voorko-
mende tolerantiefunctie waarmee de nauwkeurig-
heid van het afgeleverde resultaat gestuurd kan
worden. De specificatie kan bijvoorbeeld luiden:

“hoe kleiner de waarde van de tolerantie desto nauwkeuriger het resultaat”. Bij een betrouwbare methode zal de oplossing inderdaad nauwkeuriger worden voor afnemende waarden van de tolerantie. Het derde criterium, de efficiëntie van een numerieke oplossingsmethode, hangt nauw samen met de nauwkeurigheid die men wenst. Ik bedoel hiermee niet het vanzelfsprekende feit dat een numerieke methode duurder is naarmate men een grotere precisie eist, maar het feit dat voor hetzelfde probleem vaak methoden bestaan die efficiënt zijn als er een hoge nauwkeurigheid gevraagd wordt en nauwelijks goedkoper zijn voor lage precisies, en omgekeerd, methoden die juist voor lage precisies efficiënt zijn, echter onaanvaardbaar veel rekenwerk vergen wanneer een grote nauwkeurigheid gevraagd wordt. In een programmatheek zal men behalve over efficiënte methoden van hoge orde, ook over efficiënte methoden van lage orde van nauwkeurigheid willen beschikken, vooral ook omdat in de praktijk de wiskundige formulering van het op te lossen probleem dikwijls gepaard gaat met allerlei benaderingen en vereenvoudigingen, zodat in dergelijke gevallen het weinig zinvol is om het wiskundige model met een al te hoge precisie op te lossen.

Behalve stabiliteit, betrouwbaarheid en efficiëntie spelen ook *robuustheid en het geheugengebruik* een rol bij de keuze van geschikte oplossingsmethoden. Een methode wordt robuust genoemd, wanneer zij in staat is een grote klasse van problemen op te lossen, in tegenstelling tot de zogenaamde “special purpose” methoden die ontworpen zijn voor een kleine, wel omschreven klasse van problemen. Dergelijke specifieke methoden zijn in het algemeen veel efficiënter of vragen veel minder geheugengebruik dan de robuuste metho-

den, maar hebben het nadeel, dat de gebruiker zijn probleem nauwkeuriger dient te specificeren.

De tweede fase in de ontwikkeling van de programmatuur is de formulering van de berekeningsmethode in de vorm van een algoritme en de vertaling van de algoritme in een hogere programmeertaal, met als eindproduct een procedure. Men krijgt dan te maken met de keuze van de *programmeertaal* en de onderlinge *samenhang* van de procedures.

Voor een numerieke programmatheek dient de programmeertaal voldoende uitdrukkingsvermogen te hebben om numerieke algoritmen te beschrijven. De belangrijkste talen die aan dit criterium voldoen, zijn FORTRAN, ALGOL 60, PASCAL, SIMULA en ALGOL 68. Deze talen verschillen echter aanzienlijk in elegantie van uitdrukkingswijze. Men is het er wel over eens dat ALGOL 68 hierin duidelijk bovenaan staat; op dit moment is ALGOL 68 echter nog een taal voor fijnproevers en heeft nog een betrekkelijk geringe gebruikerskring. De talen ALGOL 60, PASCAL en SIMULA zijn veel minder geavanceerd, maar daarom makkelijker te leren en kennen dientengevolge grote groepen van gebruikers. Een nadeel van PASCAL is wel het ontbreken van dynamische array-grenzen, een ernstig gemis bij programmering van numerieke algoritmen. FORTRAN is de minst geavanceerde programmeertaal van dit vijftal. Van praktisch oogpunt uit gezien, is het echter belangrijker of er van een programmeertaal voor een voldoende aantal rekenmachines efficiënte vertalers beschikbaar zijn en hierin neemt FORTRAN zeer nadrukkelijk de eerste plaats in.

Min of meer onafhankelijk van de programmeertaal staat de onderlinge samenhang van de proce-

dures. Een procedure-bibliotheek dient niet een verzameling opzichzelfstaande procedures te vormen, maar een geïntegreerd geheel, waarin de ene procedure vaak verschillende andere procedures zal aanroepen en deze op hun beurt weer andere procedures, enz. Een dergelijke opzet maakt het mogelijk, met betrekkelijk weinig werk, zeer grote programmatheken aanzienlijk efficiënter te maken voor machines waar geen efficiënte vertalers van de gebruikte programmeertaal voor aanwezig zijn; men hoeft alleen de meest veelvuldig aangeropen procedures uit de bibliotheek in de code van de machine te vertalen en maakt daarmee deze kernstukken even zovele malen efficiënter als de machinecode efficiënter is als de programmeertaal. Een tweede facet betreft de uniformiteit van de parameterlijsten van de procedures; dit zijn lijsten, door de gebruiker in te vullen, om de procedure te kunnen besturen. Uniformiteit is in het algemeen moeilijk te handhaven: het aantal auteurs van een programmatheek is meestal groot en iedere auteur heeft zijn eigen opvattingen over de indeling van de parameterlijst. Voor de eenvoud van gebruik is uniformiteit echter van groot belang en het zou wenselijk zijn om van te voren, per gebied, een aantal parameters te specificeren waartoe de auteur zich zou moeten beperken. Deze parameters dienen probleemparameters te zijn en geen methodeparameters, dat wil zeggen bij het invullen van de parameterlijst moet men niet van de gebruiker eisen, kennis te hebben van de berekeningsmethode die aan de procedure ten grondslag ligt: de gebruiker specificeert uitsluitend zijn probleem.

3. Programmatuur voor differentiaalvergelijkingen

De ontwikkeling van numerieke programmatheken is zeker niet het voornaamste arbeidsterrein in de numerieke wiskunde. Het belangrijkste gebied waarop de numericus werkzaam is, bestaat uit het analyseren en verbeteren van bestaande berekeningsmethoden en de ontwikkeling van nieuwe methoden. Ik ben echter wel van mening, dat het uiteindelijk doel van de numericus moet zijn, direct of indirect, een bijdrage te leveren tot de ontwikkeling van steeds meer geavanceerde en complete programmatheken. Immers, alleen door middel van hoog gekwalificeerde programmatuur, kunnen de in de numerieke wiskunde behaalde resultaten toegankelijk gemaakt worden voor een breed publiek. Numerieke programmatheken zouden eigenlijk een afspiegeling, een soort momentopname moeten zijn van de bij de numerici aanwezige kennis en de nieuwe ontwikkelingen op de voet dienen te volgen. Zover zijn we nog lang niet, maar de achterstand wordt snel kleiner. Ik wil U een schets geven van de stand van zaken voor een van de belangrijkste deelgebieden in de numerieke wiskunde: de numerieke integratie van differentiaalvergelijkingen. Wanneer men de tijdschriften op het gebied van de numerieke wiskunde doorbladert, vindt men in elke aflevering wel één of meer artikelen waarin een nieuwe of verbeterde methode voor de een of andere klasse van differentiaalvergelijkingen wordt voorgesteld. Ongeveer de helft van alle publicaties in deze tijdschriften hebben de numerieke analyse van differentiaalvergelijkingen tot onderwerp. Dit gebied is nog volop in beweging en de resultaten die geboekt zijn, hebben in het algemeen nog lang niet de per-

fectie die men in andere deelgebieden van de numerieke wiskunde, bijvoorbeeld in de numerieke lineaire algebra, bereikt heeft.

Bij differentiaalvergelijkingen maakt men onderscheid tussen *randwaardeproblemen* en *beginwaardeproblemen*. Randwaardeproblemen beschrijven veelal evenwichtsproblemen uit de fysica, bijvoorbeeld de evenwichtstoestand van een rondom ingeklemd vlies waarop een niet van de tijd afhankende druk wordt uitgeoefend. Tijdsafhankelijke problemen daarentegen, worden vaak door beginwaardeproblemen beschreven; de beweging die een ingeklemd vlies uitvoert om van een gegeven begintoestand in de uiteindelijke evenwichtstoestand te komen, wordt door een beginwaardeprobleem beschreven.

3.1 Randwaardeproblemen

In de studie van randwaardeproblemen heeft men de meeste vorderingen gemaakt met betrekking tot tweekpuntsrandwaardeproblemen. Een tweekpuntsrandwaardeprobleem is de meest eenvoudige vorm van een randwaardeprobleem. Hierbij moet men niet aan een rondom ingeklemd vlies denken, maar aan een snaar die aan begin- en eindpunt ingeklemd is. De probleemstelling is dan de vorm van de snaar te vinden bij een gegeven belasting. Een reeds lange tijd bekende oplossingsmethode voor dit soort problemen, is de zogenaamde schietmethode, waarbij de randvoorwaarde in het eindpunt vervangen wordt door een additionele voorwaarde in het beginpunt. Het probleem is hiermee omgezet in een beginwaarde probleem. Men lost vervolgens dit beginwaardeprobleem op en controleert of aan de oorspronkelijke rand-

voorwaarde in het eindpunt voldaan is. De mate waarin hieraan niet voldaan is, geeft een aanwijzing hoe de additionele beginvoorwaarde aangepast moet worden. Op deze wijze tracht men iteratief de gezochte oplossing te vinden. Hoofdprobleem bij deze benaderingswijze is, dat het beginwaardeprobleem meestal instabiel is en derhalve niet nauwkeurig opgelost kan worden. Een meerbelovende techniek is de door Keller¹⁰ en Osborne¹¹ beschreven meervoudige schietmethode, waarbij het integratieinterval in meerdere deelintervallen wordt verdeeld en op ieder deelinterval een beginwaardeprobleem wordt opgelost. De benodigde additionele beginvoorwaarden worden zodanig aangepast, dat uiteindelijk een min of meer continue oplossing ontstaat die aan de randvoorwaarde in het eindpunt voldoet. Een implementatie van dergelijke technieken is nog niet beschikbaar, maar het is bekend dat een programma van Bulirsch en Stoer in een vergevorderd stadium van ontwikkeling is.

Voor randwaardeproblemen in meer dimensies is de schietmethode minder succesvol gebleken en lijkt geen serieuze gegadigde te zijn voor opname in een programmatheek.

Naast de schietmethode en haar varianten, zijn oplossingsmethoden, gebaseerd op de discretisering van de differentiaalvergelijking, belangrijk. Met behulp van de een of andere discretiserings-techniek wordt het randwaardeprobleem gereduceerd tot een stelsel, in het algemeen niet-lineaire, vergelijkingen, welke men bijvoorbeeld met een Newton-methode tracht op te lossen. De efficiëntie van deze aanpak is gekoppeld aan de efficiëntie waarmee grote stelsels lineaire vergelijkingen met zogenaamde ijle coëfficiëntenmatrices opgelost kunnen worden. Het vinden van oplossingstech-

nieken voor dergelijke ijle stelsels die gebruik maken van het ijheidspatroon van de coëfficiëntenmatrices, is een van de belangrijkste gebieden van onderzoek in de numerieke algebra.

Laten we aannemen dat het door discretisering verkregen discrete probleem met hulpmiddelen uit de numerieke algebra op te lossen is en ons nu richten op de verschillende manieren waarop een differentiaalvergelijking gediscritiseerd kan worden. De nog steeds meest gebruikte methode is gebaseerd op eindige differenties; het principe is eenvoudig: vervang het definitiegebied van de differentiaalvergelijking door een eindig aantal roosterpunten en benader alle differentiaties met op deze roosterpunten gedefinieerde differentiequotienten. Ook de uitvoering van deze wijze van discretiseren is eenvoudig zolang de roosterpunten maar op een voldoende regelmatig rooster liggen, bijvoorbeeld een rooster waarvan de mazen allen gelijk zijn. In veel problemen zal echter de oplossing in het ene gebied veel sterker variëren dan in het andere gebied, zodat men in gebieden van sterke variatie meer roosterpunten nodig heeft om de oplossing goed te kunnen representeren, dan in gebieden van geringe variatie. Een voorbeeld van zo'n meerschallig probleem vormt de klasse van singulier gestoorde tweepuntsrandwaardeproblemen welke veelvuldig optreden in de stromingsleer wanneer de stroming zogenaamde grenslagen vertoont, gebiedjes waar de snelheid zeer snel van grootte verandert. Voor dit type randwaardeproblemen is discretisering door middel van eindige differenties een verre van eenvoudig proces. Een tweede nadeel van de eindige differentiemethode is de in het algemeen minder grote nauwkeurigheid van het gediscritiseerde model in de omgeving van de randen, omdat de benadering van de

differentiaalquotienten in de randvoorwaarden veelal asymmetrische differentiequotienten impliceert.

Een discretiseringsmethode waarin niet-uniforme mazen en randvoorwaarden minder problemen opleveren, biedt de eindige elementenmethode. Deze wijze van discretiseren is een voor de computer geschikt gemaakte versie van de in de toegepaste wiskunde al lange tijd bekende en gebruikte Rayleigh-Ritz-Galerkinteknik. In de Rayleigh-Ritz-Galerkinmethode wordt niet in de eerste plaats het definitiegebied gediscretiseerd zoals bij de eindige differentiemethode, maar de oplossing zelf wordt gediscretiseerd, namelijk door deze te vervangen door een lineaire combinatie van basisfuncties. Een goed gekozen stelsel basisfuncties levert in het algemeen een nauwkeurige benadering voor de oplossing.

De eindige elementenmethode ontstond toen de computer zijn intrede deed. Wil men de Rayleigh-Ritz-Galerkinmethode implementeren, dan wordt men geconfronteerd met het feit dat het rekenen met basisfuncties alleen efficiënt verloopt wanneer deze functies uiterst eenvoudig zijn. In de eindige elementenmethode wordt dit gerealiseerd door het definitiegebied van de differentiaalvergelijking in een eindig aantal elementen – dit kunnen in principe willekeurige figuren zijn – te verdelen en op elk element een basisfunctie te definiëren die identiek nu is op alle andere elementen. Meestal kiest men polynomen omdat hiermee het eenvoudigst gerekend kan worden, maar ook rationale functies komen in aanmerking. Vermeldenswaard is verder dat een recent onderzoek van Hemker¹² uitgewezen heeft, dat het voor singulier gestoorde tweepuntsrandwaardeproblemen voordelig is exponentiële basisfuncties te kiezen.

De eindige elementenmethode heeft bewezen een zeer nauwkeurige en flexibele discretiseringsmethode te zijn, die de eindige differentiatiemethode langzamerhand van de eerste plaats lijkt te verdringen.

De verdienste de eindige elementenmethode geïntroduceerd te hebben komt de numerici niet toe. Ze is voornamelijk door ingenieurs in het leven geroepen om de zeer gecompliceerde problemen in de elasticiteitsleer en mechanica te kunnen formuleren en oplossen. Aanvankelijk werd de eindige elementenmethode ook niet herkend als een vorm van de Rayleigh-Ritz-Galerkinteknik. De onderverdeling in elementen, meestal driehoeken of rechthoeken, en de stelsels vergelijkingen voor de onbekenden werden op basis van fysische overwegingen opgesteld. Pas toen deze onbekenden geïdentificeerd werden met de gewichten van de basisfuncties in de Rayleigh-Ritz-Galerkinbenadering, werd de eindige elementenmethode erkend als een respectabele wiskundige oplossings-techniek.

Na deze uitwijding over discretiseringsmethoden, terug naar de programmatuur voor randwaardeproblemen. Reeds geruime tijd wordt door Keller¹³ en door Pereyra¹⁴ en zijn medewerkers gewerkt aan algemeen toepasbare programmatuur voor tweepuntsrandwaardeproblemen, waarbij een directe discretisering van de differentiaalvergelijking als uitgangspunt gekozen is. Voor uniforme roosters is inmiddels een programma van Lentini en Pereyra¹⁵ gepubliceerd. Een voor niet-uniforme roosters aangepaste versie van dit programma is nog in een experimenteel stadium, maar op aanvraag bij Pereyra te verkrijgen.

Voor meerdimensionale randwaardeproblemen is, afgezien van enkele op specifieke deelklassen van vergelijkingen gerichte procedures, geen programmatuur beschikbaar, hetgeen zich laat verklaren door het feit dat de automatisering van het discretisatieproces en de oplossing van het resulterende lineaire algebraprobleem aanzienlijk moeilijker zijn dan in het één-dimensionale geval.

3.2 *Beginwaardeproblemen*

Het tweede onderwerp van onderzoek op het terrein van de numerieke integratie van differentiaalvergelijkingen wordt gevormd door de beginwaardeproblemen. Ook hier weer de indeling in gewone en partiële differentiaalvergelijkingen. De numerieke analyse voor gewone differentiaalvergelijkingen is wezenlijk eenvoudiger, en de theorie is dan ook veel verder gevorderd, dan die voor partiële differentiaalvergelijkingen. De afgelopen twintig jaar heeft een ware zondvloed van publicaties te zien gegeven. Dat deze ontwikkeling toch op een redelijk geordende wijze heeft plaatsgevonden, is vooral te danken aan het theoretische werk van Dahlquist¹⁶ uit 1956 en '59. Deze publicaties van Dahlquist waren echter niet eenvoudig te lezen en het is een van de grote verdiensten van Henrici¹⁷ geweest om in zijn boeken van 1962 en 1963 het werk van Dahlquist toegankelijk gemaakt te hebben voor een grotere kring van numerici. Deze gezaghebbende en invloedrijke boeken gaven een kader aan voor de toekomstige ontwikkeling in deze tak van de numerieke wiskunde, en luiden een periode van intensief onderzoek in. De laatste jaren zijn een relatief groot aantal nieuwe boeken verschenen, waarvan enkele de plaats van

de referentiewerken van Henrici hebben ingenomen. De belangrijkste zijn het boek van Stetter¹⁸ met een functionaalanalytische benadering van de problematiek, het boek van Lambert¹⁹, minder streng dan dat van Stetter maar dan ook veel makkelijker te lezen, en tenslotte een volledig op de praktijk gericht boek van Lapidus en Seinfeld²⁰ waarin vrijwel alle op het moment van verschijnen bekende algoritmen bijeen zijn gebracht.

Bezie men de researchartikelen van de laatste jaren dan valt op dat voor de zogenaamde niet-stijve differentiaalvergelijkingen nauwelijks meer nieuwe methoden voorgesteld worden. Het accent komt langzamerhand te liggen op een steeds dieper gaande analyse van bestaande methoden. Voor stijve differentiaalvergelijkingen worden daarentegen doorlopend nieuwe integratietechnieken voorgesteld. Alvorens in te gaan op de integratie van stijve en niet-stijve vergelijkingen, een korte karakterisering van wat een stijve differentiaalvergelijking eigenlijk is. De meest kenmerkende eigenschap van dit type vergelijkingen is dat de oplossing bestaat uit zowel langzaam als snel variërende componenten waarvan de snel variërende componenten echter in absolute grootte klein zijn vergeleken bij de langzaam variërende componenten. De grafiek van de oplossing laat in het algemeen een lijn zien die aanvankelijk zeer stijl verloopt maar snel overgaat in een langzaam variërende kromme. Waar langzaam en snel verlopende processen simultaan optreden, zal de wiskundige beschrijving veelal leiden tot een stijve differentiaalvergelijking.

Alhoewel stijve differentiaalvergelijkingen veelvuldig voorkomen, werden ze pas in 1952 door Curtiss en Hirschfelder²¹ onderkend als een aparte klasse, waarvoor een bijzonder soort integratie-

techniek vereist is. Tot dan toe, en overigens ook nog lange tijd daarna, werden vrijwel uitsluitend de standaard Runge-Kuttamethode en de lineaire meerstapsmethoden van het Adams-type gebruikt, methoden die al omstreeks de eeuwwisseling bekend waren. Curtiss en Hirschfelder ontdekten dat lineaire meerstapsmethoden die geen voorgaande rechterlideovaluaties gebruiken, ook wel terugwaartse differentiatieformules genoemd, in staat zijn de stijve componenten in de oplossing van een stijve differentiaalvergelijking onder de duim te houden, waar Runge-Kutta- en Adams-methoden volledig falen.

Het belang van dit werk is de numerici echter lange tijd ontgaan. In zijn standaardwerk van 1962 merkt Henrici¹⁷ over de Curtiss-Hirschfelderformules nog op dat deze, om redenen van stabiliteit, slechts van praktische waarde zijn wanneer de orde van nauwkeurigheid niet hoger dan 6 gekozen wordt, maar dat deze formules dan altijd nog minder nauwkeurig zijn dan de corresponderende Adams-Moultonformules. Pas in 1967 voert Gear²² ze weer ten tonele door een uitvoerige analyse van de stabiliteitseigenschappen te geven, waaruit bleek dat de eerste tot en met de vijfde orde formules precies het stabiliteitsgedrag vertonen dat vereist is voor de integratie van stijve differentiaalvergelijkingen. Het artikel van Gear was tevens het begin van een stroom van publicaties op het gebied van stijve differentiaalvergelijkingen. Geen van de talloze voorgestelde integratietechnieken lijken het echter tot dusver in algemene toepasbaarheid en doeltreffendheid op te kunnen nemen tegen de terugwaartse differentiatieformules van Curtiss en Hirschfelder. Het door Gear²³ opgestelde algoritme voor deze formules, geschreven in de Nord-

sieck-representatie, waardoor variëren van integratiestap en orde van nauwkeurigheid efficiënt verloopt, maakt tegenwoordig deel uit van elke moderne programmatheek. Hiernaast treft men een aantal "special purpose" procedures aan die voordelig kunnen zijn voor specifieke problemen, maar verder geringe zelfstandige betekenis hebben. Mogelijk kunnen deze procedures wel van belang zijn als componenten in poly-algoritmen.

Voor niet-stijve beginvoorwaardeproblemen zijn een betrekkelijk groot aantal robuuste en betrouwbare programmatuurpakketten beschikbaar. Uit een omvangrijk vergelijkend onderzoek van Hull²⁴ en zijn medewerkers Enright, Fellen en Sedgwick, is gebleken dat programmatuur gebaseerd op hogere orde Runge-Kuttamethoden, extrapolatiemethoden en de Adamsformules, de beste resultaten oplevert. De Runge-Kuttamethoden en de extrapolatiemethoden, welke in wezen Runge-Kuttamethoden met variërende orde van nauwkeurigheid zijn, komen het beste tot hun recht in problemen waar de evaluatie van de rechterlidfunctie weinig rekenwerk vergt, terwijl de Adamsformules voordelig zijn wanneer een hoge nauwkeurigheid gewenst is en de rechterlidfunctie duur is. Van dit drietal zijn efficiënte implementaties, zowel in FORTRAN als in ALGOL 60, beschikbaar.

3.3 Het Noordzeeprobleem

Het laatste gedeelte van mijn betoog wil ik wijden aan het beginwaardeprobleem voor partiële differentiaalvergelijkingen. De problematiek is hier aanzienlijk moeilijker. Weliswaar zijn er numerieke technieken ontwikkeld voor beperkte, welom-

schreven probleemklassen, toch zal men tevergeefs in de huidige programmatheken zoeken naar procedures om problemen uit deze klassen te kunnen oplossen. Een van de redenen is dat de parameterlijst van zo'n procedure noodzakelijkerwijs vrij ingewikkeld is, omdat de probleemstelling zoveel gecompliceerder is dan bij gewone differentiaalvergelijkingen. Men tracht dit wel te ondervangen door een op de probleemklasse gerichte taal te ontwerpen waarmee de procedure als het ware toegesproken kan worden, maar dit soort programmatuur is nog in een experimenteel stadium en niet beschikbaar in de algemene numerieke programmatheken. In de praktijk treft men dan ook veel programma's aan die voor precies één concreet probleem geschreven zijn. Wanneer het probleem van voldoende importantie is, kan het overigens heel zinvol zijn zo'n adhoc programma als procedure in de programmatheek op te nemen. De parameter lijsten van deze procedures kunnen eenvoudig gehouden worden omdat veel van de probleemstelling in de procedure zelf ondergebracht kan worden. Een voorbeeld van een typisch adhoc, maar niet onbelangrijk probleem, is het Noordzeeprobleem. Hierin wordt gevraagd naar de waterbeweging en waterstanden tengevolge van de krachten die zon, maan en wind op het wateroppervlak uitoefenen. De wiskundige beschrijving van het Noordzeeprobleem, of wat algemener de wiskundige beschrijving van ondiep-water-problemen, leidt tot een begin-randwaardeprobleem voor een drietal gekoppelde partiële differentiaalvergelijkingen. Dat het voorspellen van waterstanden van uitzonderlijk belang is, blijkt wel uit zijn lange geschiedenis waarvan ik U een kort overzicht wil geven. Bijna tweeduizend jaar geleden spraken naturalisten als Strabo en Pliny al

het vermoeden uit dat er een verband moest bestaan tussen de hemellichamen en de beweging van de zee. Kepler realiseerde zich dat de aantrekkingskracht van zon en maan een rol speelde, maar Newton was de eerste, in 1687, die de herkomst van de getijden vanuit wiskundig oogpunt ter discussie stelde. Zich de aarde geheel overdekt denkende met water, toonde hij aan dat vooral ten gevolge van de aantrekkingskracht van de maan, het wateroppervlak een spheroïde gaat vormen. De hoofdas van deze spheroïde zou gericht zijn naar de maan en daarmee verklaarde Newton het tweedagelijks optreden van eb en vloed. Ook het verschijnsel van springtij maakte hij duidelijk aan de hand van dit model. De beschouwingen van Newton waren echter kwalitatief en ongeschikt voor praktische toepassing. In 1738 schreef de Franse Academie van Wetenschappen een wedstrijd uit voor de beste getijden-theorie. Een van de prijswinnaars was Bernoulli. Hij breidde de theorie van Newton verder uit, maar evenals Newton bracht hij de traagheidskrachten, en daarmee de beweging van de waterdeeltjes, niet in rekening. Laplace trachtte het werk van Newton en Bernoulli in dit opzicht te verbeteren. Deel IV van zijn werk *La Mécanique Céleste*, omstreeks 1800 gepubliceerd, is gewijd aan zowel een theoretische als praktische studie van de oscillaties van de zee. Zijn theorie resulteerde in algemene principes voor de samenhang tussen periodieke krachten en de daaruit voortvloeiende waterbeweging. Deze principes vormen de basis van de zogenaamde harmonische methode welke door Thomson, beter bekend als Lord Kelvin, omstreeks 1868 ontwikkeld werd en later door Darwin en anderen, begin deze eeuw, verder uitgebreid werd. Met behulp van deze methode kan het verloop van het getij

langs de kusten, uit gemeten waarden van de harmonische constanten, bij benadering berekend worden. Midden op zee is deze methode echter minder goed bruikbaar omdat de harmonische constanten, wegens het ontbreken van voldoende observatiepunten, daar niet bekend zijn. Om toch uitspraken te kunnen doen over de waterstanden in de gehele zee, dient men de partiële differentiaalvergelijkingen die de waterbeweging beschrijven, op te lossen. De belangrijkste bijdragen aan analytische oplossingsmethoden zijn geleverd door Poincaré, Taylor en Proudman, maar vooral ook, in het kader van het Deltaproject, door Van Dantzig en Lauwerier²⁵. Met analytische oplossingsmethoden kan enerzijds fundamenteel inzicht in het karakter van de getijden-beweging verkregen worden, anderzijds echter blijven de uitspraken kwalitatief, omdat nogal drastische vereenvoudigingen in het wiskundig model aangebracht moeten worden. Zo worden in vrijwel alle analytische onderzoeken de kusten met rechte lijnen benaderd, het bodemprofiel in het algemeen constant verondersteld en worden niet-lineaire termen in het model verwaarloosd. In het geval van kustwateren en ondiepe zeeën, waartoe men de Noordzee moet rekenen, is dit "rechttrekken" van kusten en bodemprofielen een te radicale vereenvoudiging. Introduceert men echter realistische kustbegrenzungen en bodemprofielen, dan laten analytische oplossingsmethoden ons in de steek en is men op numerieke methoden aangewezen. Een van de eerste numerieke onderzoeken van de partiële differentiaalvergelijkingen die de Noordzee beschrijven, dateert van 1956 en werd verricht door Hansen²⁶. Na hem hebben Fischer²⁷ in 1959, Leendertse²⁸ in 1967 en Sielecki²⁹ in 1968 meer of minder belangrijke varianten van de methode van

Hansen voorgesteld. Ook op het Mathematisch Centrum is in die tijd veel onderzoek verricht met name op het gebied van de stabiliteitsanalyse van de diverse Noordzee-schema's.

De laatste jaren is er een hernieuwde belangstelling in programmatuur voor de oplossing van ondiep-waterproblemen. Twee van de belangrijkste aanleidingen hiertoe zijn de olieboringen op de Noordzee en de nogal hachelijke situatie waarin Venetië verkeert. In beide gevallen wil men kunnen beschikken over programma's waarin de ligging van de kusten en oceaانبegrenzungen min of meer vrij gekozen kunnen worden en waarin de niet-lineaire convectieve traagheidstermen en een niet-lineaire bodemwrijvingsterm zijn opgenomen. De eerste programma's waarin dit verwezenlijkt werd, zijn de FORTRAN-programma's van Leendertse²⁸ uit 1967 en van Heaps³⁰ uit 1969. Het programma van Leendertse is gebaseerd op een impliciet schema en laat onbeperkte grote integratiestappen toe zonder gevaar voor instabiel gedrag; dat van Heaps is gebaseerd op het oorspronkelijke Hansen-schema, is expliciet en alleen stabiel voor betrekkelijk kleine integratiestappen. Anderzijds is het Hansen-schema aanzienlijk minder bewerkelijk per integratiestap dan dat van Leendertse. De vraag welke van de twee programma's het meest efficiënte is kan dus niet zonder meer beantwoord worden. Naar aanleiding hiervan is onlangs in samenwerking met studenten van de Universiteit van Amsterdam een onderzoek aangevangen, waarin een gestabiliseerd Hansen-schema en een minder bewerkelijke versie van het Leendertse-schema met elkaar vergeleken worden. In het hier geschetste geval van het Noordzee-probleem is de rechtvaardiging voor de ontwikkeling van specifieke programmatuur evident, maar er

zijn ook problemen waarvoor de ontwikkelde programmatuur zeer beperkte toepassing zal vinden, zodat het onverantwoord lijkt daar al te veel mankracht voor in te zetten. Is men wel bereid genoeg te nemen met langere rekestijden, dan kan in veel gevallen de zogenaamde semidiscretisatiemethode uitkomst bieden. In deze methode wordt het beginwaardeprobleem voor een partiële differentiaalvergelijking door middel van discretisering van de plaatsvariabelen getransformeerd tot een beginwaardeprobleem voor een stelsel gewone differentiaalvergelijkingen. Voor de discretisering komen dezelfde technieken in aanmerking die men bij discretisering van randwaardeproblemen gebruikt, namelijk de eindige differentie- en de eindige elementenmethode. Beiden leiden tot beginwaardeproblemen voor zeer grote stelsels differentiaalvergelijkingen waarin honderden, soms duizenden vergelijkingen optreden. Deze grote stelsels vereisen een geheugenzuinige en zeer stabiele integratiemethode, twee eisen die gemakkelijk met elkaar in conflict kunnen komen. Zo zullen de impliciete Curtiss-Hirschfelderformules in het algemeen wel het gewenste stabiele gedrag bezitten, maar voor problemen in meerdere plaatsvariabelen onaanvaardbaar veel geheugenplaatsen vergen. In dergelijke gevallen kunnen expliciete Runge-Kutta formules, waarin stabiliserende termen opgenomen zijn, een alternatief bieden. Het aantal benodigde geheugenplaatsen voor deze formules varieert van drie tot zesmaal het aantal te integreren differentiaalvergelijkingen. In de praktijk zal dit aantal geheugenplaatsen zelden tot problemen leiden bij verwerking op een rekenmachine.

Samenvattend mag men stellen dat zeer veel beginwaardeproblemen voor partiële differentiaalvergelijkingen bevredigend opgelost kunnen wor-

den via semidiscretisatie en met gebruikmaking van programmatuur voor gewone differentiaalvergelijkingen.

Waarde Toehoorders,

In deze rede heb ik getracht de verschillende aspecten van de vervaardiging van een numerieke programmatheek te schetsen. Vaak wordt deze laatste fase in de ontwikkeling van bruikbare numerieke wiskunde als een minder aantrekkelijke, enigszins commercieel getinte bezigheid voorgesteld. Inderdaad, het onderhoud en beheer van een programmatheek, het uitgeven van manuals, de administratie van de abonnees, dit alles heeft weinig meer met onderzoek te maken. Op de totale inspanning, nodig om tot het eindproduct te komen en dit in stand te houden, maken deze meer bedrijfsmatige werkzaamheden echter maar een gering gedeelte uit. Hoe dit verder ook zij, zonder de bouw van programmatheken zouden de vele investeringen die men in het numerieke wiskunde-onderzoek gedaan heeft, slechts ten goede komen aan de betrekkelijke kleine kring van numerici.

Mijne Heren leden van het curatorium en van het bestuur van de Stichting voor Hoger Onderwijs in de Toegepaste Wiskunde,

Voor het vertrouwen dat U in mij gesteld hebt door mij te benoemen voor deze bijzondere leerstoel ben ik U zeer erkentelijk. De openlijke aanvaarding van mijn ambt moge tevens uitdrukking geven aan mijn voornemen de mij verstrekte opdracht, luidende: "Numerieke wiskunde en informatica", naar beste vermogen te vervullen.

Geacht bestuur van deze Universiteit,

Voor de medewerking die U gegeven hebt bij mijn benoeming betuig ik U hier, bij de aanvaarding van mijn ambt, gaarne mijn dank.

Dames en Heren hoogleraren, lectoren en leden van de Wetenschappelijke Staf,

In de afgelopen jaren heeft mijn werk mij reeds met velen Uwer in contact gebracht. Het is mijn verlangen dat de mij verleende opdracht, die een nieuwe dimensie aan het door mij beoefende vak geeft, dit contact zal intensiveren.

Hooggeleerde Lauwerier,

U hebt als eerste aan een Nederlandse Universiteit de Toegepaste Wiskunde een eigen plaats gegeven. Toen in 1963 aan onze Universiteit het doctoraal-examen met hoofdrichting Toegepaste Wiskunde ingesteld werd, was dit voor mij reden om de natuurkunde vaarwel te zeggen en U als mijn mentor te kiezen. Ik beschouw het als een grote eer een van Uw eerste leerlingen geweest te zijn en bij U te hebben mogen promoveren. Onze samenwerking in het onderwijs, waarvan onze laatste caput-colleges met het gemeenschappelijke thema *Water, Wind en Wiskunde*, een voorbeeld is, stel ik ten zeerste op prijs.

Hooggeleerde van Wijngaarden,

Gaarne maak ik hier van de gelegenheid gebruik U dank te zeggen voor het vertrouwen dat U in

mij gesteld heeft, toen U de leiding van de afdeling Numerieke Wiskunde van het Mathematisch Centrum, aan mij toevertrouwde.

Hooggeachte Dekker,

Uw hartelijke ontvangst in de Vakgroep Informatica en Numerieke Wiskunde en de bijzonder prettige samenwerking die tussen ons bestaat in de diverse commissies waar wij, tengevolge van ons beider zwak voor de numerieke programmatuur, zitting in hebben, waardeer ik zeer.

Mijne Heren leden van het curatorium van het Mathematisch Centrum,

Voor Uw toestemming een onderwijstaak aan de universiteit te aanvaarden naast mijn werk op het Mathematisch Centrum, ben ik U zeer erkentelijk.

Waarde collega's van de Afdeling Numerieke Wiskunde van het Mathematisch Centrum,

Onze afdeling bestaat nu bijna vier jaar. In die vier jaar zijn de interesses van de meesten van ons geleidelijk aan komen te liggen op het gebied van de numerieke analyse van differentiaal- en integraalvergelijkingen, en van de ontwikkeling van numerieke programmatuur. De plezierige verstandhouding waarin wij als team werken betekent veel voor mij.

Dames en Heren studenten,

Het is mijn voornemen U niet alleen wegwijs te maken in de vele numerieke technieken, maar

vooral ook samen met U onderzoek te doen op de
nog vele braakliggende terreinen van de nume-
rieke wiskunde.

Dames en Heren,

Ik dank U allen voor Uw aandacht.

NOTITIES EN REFERENTIES

- 1 J. R. Rice, *Mathematical Software*, Academic Press, p. 4, 1971.
- 2 F. L. Alt, "Multiplication of Matrices", *Mathematical Tables and other Aids to Computation*, 2, p. 12-13 (1946).
- 3 P. Naur (ed.), "Revised Report on the Algorithmic Language ALGOL 60", *Regnecentralen, Kopenhagen* (1962).
- 4 Een introductie tot de taal PASCAL en het volledige Revised Report van Standard PASCAL treft men aan in: Kathleen Jensen en N. Wirth, *PASCAL, User Manual and Report*, Lecture Notes in Computer Science, 18, Springer Verlag 1974.
- 5 O.-J. Dahl, B. Myhrhaug en K. Nygaard, "SIMULA 67, Common Base Language", *Norwegian Computing Centre, Oslo* (1967).
- 6 A. van Wijngaarden (ed.), "Report on the Algorithmic Language ALGOL 68", *Numerische Mathematik*, 14, p. 79-218 (1969).
- 7 *Collected Algorithms from Communications of the Association for Computing Machinery*, Association for Computing Machinery. New York.
- 8 J. H. Wilkinson en C. H. Reinsch, *Handbook for Automatic Computation, II Linear Algebra*, Springer Verlag 1971.
- 9 De belangrijkste numerieke programmatheken die bij het Academisch Rekencentrum in Amsterdam beschikbaar zijn, zijn de FORTRAN-programmathiek IMSL ontwikkeld door de *International Mathematical and Statistical Libraries, Inc.* en verkrijgbaar bij IMSL, Sixth Floor GNB Building, 7500 Bellaire Boulevard, Houston, Texas 77036, USA, de programmatheek NAG, zowel in FORTRAN als ALGOL 60, ontwikkeld door de *Numerical Algorithm Group* (de vroegere *Nottingham Algorithm Group*) en verkrijgbaar bij NAG, Central Office, Oxford University Computing Laboratory, 13 Banbury Road, Oxford OX 26 NN, U.K., en tenslotte de ALGOL 60- en PASCAL-programmatheken, NUMAL en NUMPAS ontwikkeld door de *Afdeling Numerieke Wiskunde van het Mathematisch Centrum*.
- 10 H. B. Keller, *Numerical Methods for Two-point Boundary Value Problems*, Blaisdell 1968.
- 11 M. R. Osborne, "On Shooting Methods for Boundary

- Value Problems", *J. Math. Anal. Appl.*, **27**, p. 417-433 (1969).
- 12 P. W. Hemker, *A Numerical Study of Stiff Two-point Boundary Value Problems*, te verschijnen in de MCT-reeks van het Mathematisch Centrum te Amsterdam.
 - 13 H. B. Keller, "Accurate Difference Methods for Non-linear Two-point Boundary Value Problems", *SIAM J., Numer. Anal.* **11**, p. 305-320 (1974).
 - 14 V. Pereyra en E. G. Sewell, "Mesh Selection for Discrete Solution of Boundary Value Problems in Ordinary Differential Equations", *Numer. Math.* **23**, p. 261-268 (1975).
 - 15 M. Lentini en V. Pereyra, "A Variable Order Finite Difference Method for Non-linear Multipoint Boundary Value Problems", *Math. of Comp.* **28**, p. 981-1003 (1974).
 - 16 G. Dahlquist, "Convergence and Stability in the Numerical Integration of Ordinary Differential Equations", *Math. Scand.* **4**, p. 33-53 (1956).
—, "Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations", *Kungl. Tekniska Hogskolans Handlingar*, **130** (1959).
 - 17 P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley 1962.
 - 18 H. J. Stetter, *Analysis of Discretization Methods for Ordinary Differential Equations*, Springer Verlag 1973.
 - 19 J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley 1973.
 - 20 L. Lapidus en J. H. Seinfeld, *Numerical Solution of Ordinary Differential Equations*, Academic Press 1971.
 - 21 C. F. Curtiss en J. O. Hirschfelder, "Integration of Stiff Equations", *Proc. Nat. Acad. Sci. U.S.A.*, **38**, 235-243 (1952).
 - 22 C. W. Gear, "Numerical Integration of Stiff Ordinary Differential Equations", *University of Illinois Dept. of Computer Science, Report 221* (1967).
 - 23 C. W. Gear, "Algorithm 407, DIFSUB for Solution of Ordinary Differential Equations", *Comm. ACM.*, **14**, 185-190 (1971).
 - 24 T. E. Hull, W. H. Enright, B. M. Fellen en A. E. Sedgwick, "Comparing Numerical Methods for Ordinary Differential Equations", *SIAM J., Numer. Anal.* **9**, p. 603-637 (1972).
 - 25 Bijdragen II.4 en II.5 in het *Rapport Deltacommissie*, deel 3, Staatsdrukkerij- en Uitgeversbedrijf, 's-Gravenhage (1960).
 - 26 W. Hansen, "Theorie zur Errechnung des Wasserstandes

- und der Strömungen in Randmeeren nebst Anwendungen", *Tellus* **8**, p. 287-300 (1956).
- 27 G. Fischer, "Ein numerisches Verfahren zur Errechnung von Windstau und Gezeiten in Randmeeren", *Tellus* **11** (1959).
 - 28 J. J. Leendertse, "Aspects of a Computational Model for Long-period Waterwave Propagation", *Rand Memorandum* RM-5294-PR, *The Rand Corporation*, Santa Monica (1967).
 - 29 A. Sielecki, "A Numerical Method for the shallow water equation", *Mon. Weath. Rev. U.S. Dep. Agric.*, **96**, p' 150-156 (1968).
 30. N. S. Heaps, "A Two-dimensional Numerical Sea Model" *Philos. Trans. Roy. Soc. London Ser. A* **265**, p. 93-137 (1969).

