

3.0 SEP. 1959

Verslag van de internationale conferentie over informatie
verwerkende machines te Parijs, juni 1959

door Dr. B. Heyna

Deze conferentie was georganiseerd door de U.N.E.S.C.O. met het doel het onderzoek op het gebied van elektronische rekenmachines te coördineren en tot uitwisseling te komen van resultaten die in verschillende landen (min of meer los van elkaar) verkregen zijn. Dat dit nuttig en nodig was, is tijdens de vergadering wel gebleken, waar naar voren is gekomen dat sommige problemen tegelijkertijd, en op vrijwel dezelfde wijze, opgelost zijn in de Angelsaksische landen enerzijds en Rusland anderzijds.

De conferentie werd gehouden van 15 - 20 juni 1959 in het gebouw van de U.N.E.S.C.O. te Parijs met uitzondering van de openingszitting, die plaats vond in de "Sorbonne", de bekende universiteit van Parijs. In tegenstelling echter tot het mathematisch congres in Amsterdam in 1954, waar de belangrijkste lezing tijdens de openingszitting werd gehouden, werden hier alleen enkele algemeenheden medegedeeld (naast welkomstwoorden etc.). Zo men dit al als een tegenvaller zou willen noteren, dan is die door de rest van de conferentie meer dan goedge maakt. Tegelijkertijd was nog een tentoonstelling georganiseerd: de "Automath" waar een aantal elektronische rekenmachines door hun betreffende fabrikanten geëxposeerd werd. Behalve bekende firma's als I.B.M. en Bull, trof ik hier ook enkele japanse firma's aan, die machines demonstreerden waarin een nieuw bouw-element was gebruikt: de "parametron", een japanse vinding.

De organisatie van de conferentie was uitstekend. Hoewel er ruim 130 voordrachten zijn gehouden, is de tijd, aangegeven in het programma, vrijwel nooit overschreden. Verder werd aan iedere deelnemer bij het begin van de conferentie een lijst met "abstracts" uitgereikt, zodat ieder van te voren er al over kon denken wat hij eventueel te vragen had. De vragen over een bepaalde lezing moesten vóór afloop van die lezing schriftelijk bij de voorzitter ervan ingediend worden, zodat deze kon beoordelen hoeveel tijd iedere vragensteller ter beschikking had en eventueel kon beslissen dat bepaalde vragen zouden vervallen.

Tot het slagen van de conferentie heeft ook de inrichting van het gebouw van de U.N.E.S.C.O. bijgedragen. De vergaderzalen waren "airconditioned" zodat het binnen aangenaam koel was, ondanks de hitte buiten. Verder was er een vertaalinstallatie, zodat iedere voordracht naar vertaling beluisterd kon worden in het engels, frans of russisch. Verder waren in de hal een aantal stoelen en banken geplaatst, zodat hier een ongezochte gelegenheid bestond om met de andere deelnemers over bepaalde onderwerpen nog eens van gedachten te wisselen. Tevens was nog een zaaltje ter beschikking van de deelnemers gesteld, waar deze de gelegenheid hadden enkele onderwerpen nog eens te overdenken of notities uit te werken.

Aan het begin van iedere dag werd aan de deelnemers een krantje verstrekt, met het programma van die dag en de nog volgende dagen. Tevens werden hierin andere mededelingen gedaan die voor de deelnemers van belang waren.

De voordrachten

Er waren ruim 130 lezingen, waarvan er 61 werden voorgedragen in zgn. "plenary sessions", die elk een middag of een morgen in beslag namen (zie bijgevoegd programma), verder waren er 65 voordrachten tijdens colloquia, die 's middags gehouden werden en nog een vijftal avondlezingen. Deze laatste stonden op een iets ander niveau dan de overige lezingen, niettemin vond ik ze toch wel aardig om te beluisteren.

De lezingen voor de "plenary sessions" waren onderverdeeld in de volgende groepen:

- a. Methoden van digitaal rekenen.
- b. Logische opbouw van digitale rekenmachines.
- c. Genormaliseerde symbolische taal voor digitale rekenmachines.
- d. Automatische vertaling.
- e. Lezen en herkennen van tekens of signalen en leren door machines.
- f. De toekomstige technische ontwikkeling van de rekenmachines.

De colloquia handelden over de volgende onderwerpen:

1. Het terugvinden van informatie bij zeer grote geheugencapaciteiten.
2. Schakelalgebra.
3. Het verband tussen digitale en analoge rekenmachines.

4. De logische constructie van zeer kleine rekenmachines.
5. Lineaire programmering.
6. Fouten ontdekken en corrigeren.
7. Verzamelen, bewaren en terugvinden van informatie.
8. Machinale vertaling.
9. Automatisch programmeren.
10. Logische constructie van zeer snelle rekenmachines.
11. Numerieke analyse (= algebra) door rekenmachines.
12. Methoden ter oplossing van lineair systemen.

Aangezien elke middag naast de "plenary session" nog 2 of 3 colloquia werden gehouden, kon ik niet bij alle lezingen aanwezig zijn. Ik heb gekozen de colloquia no. 3 en 11, op dinsdag- resp. vrijdagmiddag, voor de overige tijd heb ik de "plenary sessions" bijgewoond.

Over de inhoud van de lezingen wil ik nog het volgende opmerken.

groep a

De onderwerpen uit de numerieke wiskunde, die bij het K.N.M.I. het meest voorkomen, nl. (partiële) differentiaal-vergelijkingen (Noordzee, numerieke weersverwachting) en matrices (methode van de kleinste kwadraten etc.), bleken ook de hoofdschotel te vormen van de lezingen die tot groep a behoorden.

Speciaal wat betreft de partiële differentiaal-vergelijkingen geldt dat tot voor kort (d.w.z. voor de invoering van elektronische rekenmachines) er vrijwel geen methoden bekend waren om deze vergelijkingen op te lossen. Dit vond zijn oorzaak in het feit dat in bijna alle gevallen geen analytische oplossing mogelijk is, terwijl numerieke methoden met de toen bestaande hulpmiddelen zó bewerkelijk waren, dat dit hun toepassing uitsloot. Met de invoering van de elektronische rekenmachines ontstond hier een soort vacuum: de mathematici konden nu wel beschikken over de benodigde rekencapaciteit, maar de methoden om een partiële differentiaal-vergelijking om te zetten in een reeks elementaire bewerkingen (optellen, aftrekken, vermenigvuldigen enz.) ontbraken of waren slechts in rudimentaire vorm aanwezig. Vandaar dat deze problemen in de laatste jaren van verschillende kanten met kracht zijn aangepakt, waarbij duplicaties niet altijd voorkomen konden worden.

Enkele van de belangrijkste aspecten die over dit onderwerp naar

voren kwamen zijn wel de iteratie-methoden en de afrondingsfouten. Afrondingsfouten ontstaan doordat men transcendente functies in een reeks ontwikkelt en deze reeks na een zeker aantal termen afkapt, verder door het vervangen van differentiaal uitdrukkingen door eindige differenties en ten slotte tijdens de berekening. Wanneer men twee getallen, elk van n cijfers, met elkaar vermenigvuldigt, verkrijgt men een getal van $2n$ of $2n-1$ cijfers, wanneer men dit niet afrondt, zou in een praktisch probleem, tengevolge van de vele vermenigvuldigingen, de lengte van de getallen al snel buiten alle proporties groeien. Men moet dus telkens afronden; per keer betekent dit slechts een kleine fout, maar door hun grote aantal kunnen deze fouten tezamen toch een aanzienlijk bedrag vormen.

Het vinden van een absolute grens voor de fout in het eindresultaat levert dikwijls grote moeilijkheden. Bovendien geeft dit een bedrag, dat slechts in zeer uitzonderlijke gevallen bereikt zal worden (nl. als van alle tussenresultaten de fouten dezelfde kant uit werken). Beter is daarom de afrondingsfouten als toevallige grootheden te beschouwen en de fout in het eindresultaat op statistische manier te bepalen. Dit is op theoretische gronden verdedigd o.a. door Blanc, terwijl het door Henrici in de praktijk getoetst is door enkele eenvoudige differentiaal-vergelijkingen, waarvan een analytische oplossing bekend was, op numerieke wijze door een elektronische rekenmachine te laten oplossen. De fouten bleken inderdaad aan de verwachte statistische wetten te voldoen.

Nauw verbonden met het schatten van de fout is de controle van het resultaat. Wanneer men, bv., een matrix heeft geïnverteerd, dan kan men de uitkomst vermenigvuldigen met de oorspronkelijke matrix. Er moet dan de eenheidsmatrix uitkomen, hetgeen men door de machine zelf kan laten testen. Tengevolge van de afrondingsfouten zullen er echter kleine verschillen kunnen optreden. Om de machine een instructie te kunnen geven welke verschillen nog toelaatbaar zijn, is een studie van de afrondingsfouten noodzakelijk.

Iteratieve processen

Hieronder verstaat men het bepalen van een reeks successievelijke benaderingen. Deze methode is bijzonder geschikt voor matrices en elliptische differentiaal-vergelijkingen. Als voorbeeld nemen we de eenvoudigste elliptische differentiaal-vergelijking, de Laplace vergelijking:

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = 0 \quad (\text{met gegeven randwaarden}) \quad (1)$$

Door invoeren van eindige differenties kan deze vergelijking omgevormd worden tot:

$$\vec{z} = T\vec{z} + \vec{b} \quad (2)$$

Hierin is \vec{b} een vector (bepaald door de randwaarden), T een matrix en \vec{z} een vector, waarvan de componenten gelijk zijn aan de waarden van z in de roosterpunten van het net, dat gebruikt is bij de invoering van de eindige differenties. We kiezen een willekeurige beginvector $\vec{z}^{(0)}$ en bepalen achtereenvolgens $\vec{z}^{(1)}$, $\vec{z}^{(2)}$... $\vec{z}^{(n)}$ uit:

$$\vec{z}^{(n)} = T \vec{z}^{(n-1)} + \vec{b} \quad (3)$$

Dit procédé is zeer geschikt voor elektronische rekenmachines. Er zitten echter nog wel enkele problemen aan vast, zoals bv. dat van de convergentie. Zoals de Rus Dorodnicyn opmerkte, is hierbij de praktische convergentie belangrijker dan de theoretische:

Een reeks die in een klein aantal stappen zeer dichtbij het gewenste resultaat komt, om zich er daarna weer van te verwijderen is soms beter geschikt dan een andere, die theoretisch wel convergeert, maar waarbij een groot aantal stappen nodig is om de vereiste nauwkeurigheid te bereiken.

Een aantal andere sprekers hield zich bezig met methoden om het aantal iteraties, nodig om een bepaalde nauwkeurigheid te verkrijgen, te verminderen. Dit is een praktisch belang, het spaart (dure) machinetijd.

groep b

In de ontwikkeling van elektronische rekenmachines kan men twee facetten onderscheiden, nl. ten eerste het verbeteren van de bouwstenen (electronenbuizen, transistors, ferrietkernen enz.) en ten tweede het verbeteren van de wijze waarop deze bouwstenen samengevoegd worden tot één geheel. Het tweede facet, dat men aanduidt met de naam logische opbouw ("logical design") was het onderwerp van de lezingen uit groep b. Een eenvoudig voorbeeld, afkomstig van achter het ijzeren gordijn, kan de bedoeling van dit onderwerp wellicht verduidelijken. Het vermenigvuldigen in het tweetalig stelsel geschiedt als regel door herhaalde optellingen gescheiden door een verschuiving van de vermenigvuldiger a.v.

tweetallig	(10-tallen)
1010001101	(653)
1001110111	(631)
<hr/>	<hr/>
1010001101	
1010001101	
1010001101	
1010001101.	
1010001101	
1010001101	
1010001101..	
<hr/>	<hr/>
1100100100110001011	(412043)

De machine tast de vermenigvuldiger bit voor bit af ("bit" is de afkorting voor "binary digit"), te beginnen rechts. Is een bit = 1, dan wordt het vermenigvuldigtal opgeteld en één plaats naar links verschoven. Is een bit = 0, dan wordt alleen naar links verschoven.

Voor bovenstaande vermenigvuldiging zijn dan 9 verschuivingen en 7 optellingen nodig, die elk een bepaalde tijd nodig hebben. Men kan de totaaltijd verkorten door de vermenigvuldiger a.v. te schrijven:

$$101000\bar{1}00\bar{1}$$

waarbij $\bar{1}$ een negatieve bit betekent, aangevende dat de machine moet aftrekken i.p.v. optellen (de tijd voor een aftrekking is even groot als voor een optelling). Verder kan men de machine zo schakelen, dat de vermenigvuldiger in één keer over een aantal plaatsen schuift. De totale bewerkingstijd wordt dan in het bovenstaande geval 4 opteltijden + 3 verschuiftijden (tegen 7 + 9 in het voorgaande geval).

Andere onderzoekers proberen de tijd, nodig voor de tweetallen-overdracht te verminderen. Deze tijd neemt nl. bij parallel optellen, d.i. de vlugste manier van optellen, verreweg het grootste deel van de totale opteltijd in beslag. Hierbij worden verschillende wegen gevolgd; m.i. is de ideale methode nog niet gevonden.

Ook op een ander niveau is men druk bezig met het zoeken naar verbeteringen. De bovengenoemde bouwstenen worden nl. samengevoegd tot min of meer los van elkaar staande organen (rekenorgaan, geheugen enz.). Hierbij treedt het probleem van de verschillende snelheden op. Het rekenorgaan is b.v. veel sneller dan de in- en uitvoerorganen en de moeilijkheid is nu om ervoor te zorgen dat een snel orgaan niet het grootste deel van de tijd onbenut blijft, omdat het staat te wachten op een langzaam orgaan. Door verschillende onderzoekers wordt op verschillende wijze getracht om dit

probleem op te lossen. Zo kan men verschillende leesapparaten (al dan niet onderling aan elkaar gelijk) parallel schakelen, die ieder een deel van de benodigde informatie lezen. In de gamma 60 is men zover gegaan, dat de diverse organen zelfstandige eenheden vormen, een centraal orgaan, de director, verdeelt de instructies en de informatie over de diverse organen, waarbij zekere voorrangregels in acht genomen worden. Dit gebeurt hier dus automatisch, bij andere (iets eenvoudiger) machines moet de programmeur er voor zorgen. Ook laat men wel twee problemen (waarvan het ene bv. slechts $3/4$ van het geheugen nodig heeft en het ander $1/4$) tegelijkertijd door de machine berekenen, waardoor voorkomen wordt dat een deel van het (kostbare!) geheugen tijdelijk ongebruikt blijft. Leiner (U.S.A.) gaat zelfs zo ver om drie geheel verschillende rekenmachines aan elkaar te koppelen, waarbij de ene zeer snel rekent, de tweede bijzonder geschikt is voor het in- en uitvoeren van gegevens en de derde een "general purpose" machine is, die tegelijk de werkzaamheden van de andere twee regelt.

Zoals alles in deze paragraaf uit twee bestaat, werken ook de resultaten van de onderzoeken in twee richtingen: enerzijds de constructie van steeds snellere en grotere (maar ook duurdere machines), anderzijds - door de gevonden vereenvoudigingen - naar steeds economischer machines die weliswaar een bescheiden omvang hebben, maar evenveel presteren als vroeger alleen door duurdere machines mogelijk was. Ik wil dan ook het overzicht over deze groep besluiten met een verwijzing naar de lezing van Lehman (Israël) die claimt een machine (vergelijkbaar met bv. de Zebra) te kunnen vervaardigen door zelfbouw voor \$ 12.000,--, alles inbegrepen (dus ook arbeidslonen).

groep d

Met automatische vertaling wordt bedoeld, dat een stuk tekst in een bepaalde taal op een of andere manier aan de machine toegevoerd wordt en er dan zonder verder menselijke tussenkomst in een andere taal weer uitkomt. Tot voor kort hielden taaldeskundigen het voor onmogelijk, dat men hierbij verder zou komen dan een soort woord voor woord vertalen. De lezingen uit groep d toonden aan dat laatstgenoemde zienswijze niet juist is. Hierbij dient evenwel vermeld te worden, dat de structuur van een stuk wetenschappelijke tekst i.h.a. meer geschikt is voor machinale vertaling, dan bv. een literair kunstproduct.

De onderzochte vertalingen waren russisch-engels, engels-russisch en engels-japans. De machines bleken daarbij o.m. tot het volgende in staat: opzoeken van de stam van een woord, zinsontleding en toepassen van de grammatica, zowel voor de oorspronkelijke als voor de vertaalde tekst. Het programmeren van de regels van de grammatica bleek daarbij niet zo zeer moeilijkheden op te leveren vanuit machine technisch oogpunt, dan wel vanwege het feit dat deze regels door de taalgeleerden nog niet voldoende onderzocht waren, althans niet op mathematische manier. Vandaar dan ook dat men in de Rand Corporation (Californië) is begonnen met het opbouwen van een eigen grammatica. Niet dat deze in strijd is met de officiële grammatica, maar hij levert het antwoord op een aantal vragen die in laatstgenoemde nog open zijn. Bv. als een zelfstandig naamwoord twee betekenissen heeft, welke bijvoegelijke naamwoorden komen dan uitsluitend voor in combinatie met dit zelfstandig naamwoord in de eerste betekenis, en welke in de tweede betekenis, of: hoe vaak en bij welke combinaties van twee woorden moet de volgorde bij vertaling omgewisseld worden enz.

Intussen heeft men op de Harvard Universiteit ook onderzoeken gedaan met de veel verguisde woord-voor-woord vertaling (van russisch in engels). Hoewel de vertaling een H.B.S.-leerling een dikke onvoldoende zou bezorgen, was hij toch beslist leesbaar, dus positief van waarde voor iemand die wel engels, maar geen russisch kent.

groep e

Een van de "bottleneck's" bij het machinaal verwerken van informatie is het overbrengen van de informatie vanuit de oorspronkelijke vorm in een vorm die geschikt is voor machinale bewerking. Er zijn al enkele pogingen in die richting gedaan (photolecture, marksensing) maar deze vereisen nog speciale bewerkingen. Het zou van groot economisch voordeel zijn (ook voor het K.N.M.I.) als men geschreven tekst direct aan de machine kon toevoeren, zonder eerst ponskaarten, ponsband of dergelijke te behoeven te maken.

Bij het lezen van geschreven tekst moet men onderscheid maken tussen gedrukte, getypte en (met de hand) geschreven tekst. Eerstgenoemde is het meest uniform, bij met een schrijfmachine getikte tekst is de vorm van de letters wel uniform, maar er treden vaak afwijkingen op wat de zwarting betreft (de letter e bijvoorbeeld kan opgevuld zijn, geeft verwarring met o; of de horizontale streep kan erg flauw zijn, geeft verwarring met c).

In Japan heeft men het probleem van het lezen van schrijfmachineschrift a.v. aangepakt: Een rechthoek die zo groot is dat alle schrijfmachinetekens er in passen, wordt verdeeld in $10 \times 12 = 120$ vierkantjes, die door een lichtstraal rij voor rij worden afgetast (dus ongeveer als het beeld bij televisie). Is de terugkaatsing boven een bepaalde drempel, dan wordt het vakje als wit gekenmerkt, onder die drempel als zwart. Iedere letter (of cijfer) wordt dus als een bepaald patroon van witte en zwarte vierkantjes weergegeven, de machine moet uit de veelheid van patronen 73 groepen kunnen onderscheiden, die elk een hoofdletter, kleine letter, cijfer of een leesteken voorstellen. Verschillende moeilijkheden moesten daarbij nog overwonnen worden (bv. horizontale of verticale verschuiving van de letters). Bij de eerste testproef bleek de machine 70% van de letters goed uit te typen (en dus goed gelezen te hebben), het falen der resterende 30% lag voornamelijk aan het niet goed gecentreerd zijn van de letters in het beeldvlak. Men is bezig dit te verbeteren.

Bij het M.I.T. in de Verenigde Staten werkt men aan een procédé om geschreven letters (en cijfers) door de machine te laten lezen; het beeldvlak wordt hier in ca. 1000 rechthoekjes verdeeld. Het herkennen geschiedt hier op grond van topologische gegevens: aantal dubbelpunten, gesloten of open kromme enz. Bij een eerste proef werden van 120 letters er 83 goed gelezen, 6 fout en 31 kon de machine niet lezen. Gezien de vorm van de te lezen letters (waarvan er een aantal d.m.v. een lantarenplaatje getoond werden) vond ik dit een zeer goed resultaat.

Een afwijkende methode werd gevolgd door Ganzhorn (Duitsland) die de omtrek van cijfers liet volgen door een lichtstraal, waarbij het aftastmechanisme een elektrische spanning levert die evenredig is aan de afgeleide (naar de tijd) van de uitwijking van de lichtstraal. De toppen en tekenwisselingen van deze spanning vertonen een patroon dat karakteristiek is voor het bijbehorende cijfer, en dat hetzelfde is voor grote en kleine, scheve en rechte en horizontaal of verticaal verschoven cijfers. Voor met de hand geschreven cijfers is dit een belangrijk voordeel.

Een originele methode is ontwikkeld door de technische hogeschool in Karlsruhe (Duitsland). Door middel van een aftaster, fotocel en diodes wordt in een weerstandsnetwerk een potentiaalverdeling teweeggebracht, zodanig dat elk punt corresponderende met een punt van de te lezen letter op potentiaal E is, en de rand van het weerstandsnet (corresponderende met de rand van het beeldvlak) op potentiaal nul. In tegenstelling tot andere methoden, waar de informatie betreffende een bepaald punt van het

beeldvlak alleen maar luidt: zwart of wit, geeft het potentiaalveld veel meer informatie, ook over de omgeving van het referentiepunt, bv.:

1. Potentiaal van het weerstandsnet = E , hoewel de scanner een potentiaal nul geeft. Dit betekent dat het referentiepunt door een gesloten of bijna gesloten kromme omgeven is.
2. 1e afgeleide positief, 2e afg. = 0: Een (bijna) rechte lijn rechts van het referentiepunt.
3. 1e afgeleide positief, 2e afg. negatief: gekromde lijn, open naar links.
4. (af te leiden uit de stroomverdeling, die uiteraard ook gemeten kan worden): sterke concentratie van stroomlijnen: scherpe punt.

Een ander voordeel van deze methode is, dat de letters automatisch gecentraliseerd kunnen worden. Staat een letter bv. teveel naar links in het beeldvlak, dan gaat er door de rechterkant van het weerstandsnet minder stroom dan door de linker, dit stroomverschil kan gebruikt worden om de scanner te justeren (d.m.v. een automatische terugkoppeling).

Persoonlijk ben ik van mening, dat deze methode het meest aansluit bij het menselijke lezen, doordat krommingen naar rechts of naar links en hun onderlinge samenhang onderzocht kunnen worden.

Een nog moeilijker probleem is om de machine gesproken woorden te laten herkennen. Hierover was één lezing, door Halle van het M.I.T. in de V.S. De moeilijkheden zijn vele, maar principieel niet onoplosbaar. Men is er in geslaagd 70 los van elkaar staande woorden, elk uitgesproken door 15 verschillende personen, door de machine te laten identificeren. Hiermee is bv. de mogelijkheid geschapen om de machine te laten werken op commando's: start, send, cease, keep, two, read one, add up etc. Overigens is dit nog maar een partiële oplossing van het probleem. Indien de volledige oplossing lukt, zijn er veel meer en belangrijker toepassingen.

Uit het voorgaande blijkt dat de machine steeds meer menselijke handelingen kan verrichten: aanvankelijk alleen rekenen en onthouden van getallen, later ook onthouden en eventueel zelf veranderen van opdrachten, en binnenkort wellicht lezen, luisteren en vertalen. De vraag ligt voor de hand: kan de machine ook iets leren? Bij het woord "leren" wordt daarbij niet in de eerste plaats gedacht aan uit het hoofd leren. Het is duidelijk dat de machine een rijtje jaartallen, of de namen van de presidenten van de V.S. in

principe voor onbepaalde tijd onthouden kan als deze eenmaal in het geheugen gebracht zijn. Wel is men geïnteresseerd in leren door ondervinding, of leren door onderwijzen. Dit blijkt inderdaad mogelijk, zij het voorlopig nog op elementair niveau. Wanneer er bv. 3 verschillende methoden zijn om een bepaald type differentiaalvergelijking op te lossen, dan kan men de machine zelf laten beslissen welke der 3 methoden hij zal kiezen (eventueel in afhankelijkheid van de beginvoorwaarden), wanneer hij reeds een aantal van deze vergelijkingen heeft opgelost en aldus de genoemde methoden heeft kunnen testen op convergentie, nauwkeurigheid enz.

groep f

De inleider van deze lezingen deelde mee, dat de lezingen uit deze groep projecten behandelden, die nu in studie zijn, en waarvan men mag verwachten dat ze binnen 3 à 5 jaar op de markt zullen komen. Of dit inderdaad voor alle projecten uit deze groep bewaarheid zal worden, waag ik te betwijfelen, niettemin was deze gelegenheid om een kijkje achter de schermen te nemen zeer interessant. Dat de ontwikkeling gaat in de richting van nog snellere en nog grotere machines was niet zo verbazingwekkend. Wel werd ik getroffen door de getallen die men noemde: opteltijden van 20 nanosec (1 nanosec = 10^{-9} sec) en geheugens van 10^8 bits, d.i. ca. 2.000.000 getallen van 12 (digitale) cijfers. Ter vergelijking diene, dat men voor de snelste bestaande machines (waaronder ik die machines versta waarvan er een aantal geleverd zijn en die hun bruikbaarheid in de praktijk bewezen hebben) bovengenoemde getallen mag stellen op 20 microsec (d.i. 20×10^{-6} sec) en ca. 20.000 getallen.

Dat voor het bereiken van dergelijke mabitieuze getallen nieuwe technieken ontwikkeld moeten worden, laat zich begrijpen. Een opteltijd van 20 n.sec vereist een herhalingsfrequentie van de afzonderlijke impulsen van minstens 1000 MHz, hetgeen in vacuum een golflengte van 30 cm betekent, en in vaste stof nog minder. Dit betekent dat de verbindingen in deze (toekomstige) machine aanzienlijk kleiner moeten zijn dan 30 cm, zodat de machine zelf niet veel groter kan zijn dan een gewone (niet elektronische) tafelrekenmachine, of zelfs niet groter dan een sigarendoosje! In het laatste geval stoppen we hem meteen maar in een Dewar-vat. Ittner (U.S.A.) hield nl. een lezing over de mogelijkheden van een rekenmachine waarbij gebruik gemaakt wordt van suprageleiding van materie bij lage temperatuur. De technische moeilijkheden zijn echter nog vele.

Aanmerkelijk dichter bij een praktische verwezenlijking zijn de pogingen van Proebster (Zwitserland), die een dunne film gebruikt van 80% Ni en 20% Fe, in vacuum opgedampt bij aanwezigheid van een magnetisch veld op glas van 0,1 mm dikte. Zulke films blijken merkwaardige magnetische eigenschappen te hebben, die fysisch nog niet geheel verklaard zijn, maar waarschijnlijk samenhangen met het feit dat de dikte van de film (i.c. 0,0001 mm) veel kleiner is dan de gemiddelde afmeting van een gebiedje van Weiss. Men heeft een experimentele matrix van 32 x 10 bits geconstrueerd, die zowel wat snelheid als aantal bits per mm³ een orde van grootte beter is dan wat bereikbaar is met een matrix van ferrietkernen. Theorie, zowel als experiment duiden er echter op, dat men beide grootheden nog wel met een factor 10 tot 100 zal kunnen verbeteren. Hiervoor moeten dan nog wel enkele technische moeilijkheden opgelost worden; bv. het aanbrennen van een stel evenwijdige draden op onderlinge afstanden van 0,01 mm. Met de gewone bedradingstechniek is dit onmogelijk, maar het procédé van gedrukte schakeling ("printed circuits") biedt hier mogelijkheden. Men zoekt het in een soort sandwich structuur, waarbij afwisselend geleidende, isolerende en halfgeleidende lagen worden opgedampt, waarbij tevens de benodigde diodes en transistoren geschapen worden (vandaar de halfgeleidende lagen). Een voordeel van deze methode is, dat de magnetische elementjes en transistoren nu niet meer stuk voor stuk gefabriceerd en gemonteerd behoeven te worden, hetgeen een aanmerkelijke kostenbesparing met zich brengt.

Colloquium 3

In tegenstelling tot enige tijd geleden, toen analoge- en digitale rekenmachines vrijwel niets met elkaar gemeen hadden, treft men nu in sommige digitale machines ook analoge onderdelen aan (of omgekeerd). Dit levert voor bepaalde problemen besliste voordelen op, genoemd werden: microseismen, frequentieanalyse, autocorrelatie (Braffort). Wischmeyer wees op de overeenkomsten die er bestaan tussen het bepalen van correlatie en convolutie. Hij had zijn formules van te voren vastgelegd in lantarenplaatjes, waardoor hij in de hem toegestane tijd aanzienlijk meer kon vertellen dan zonder dit trucje mogelijk zou zijn geweest, maar waardoor ik geen tijd had om de formules over te nemen. Sommige slimmerikken hadden hier blijkbaar op gerekend en een fototoestel meegenomen.

Over het omzetten van een analoge veranderlijke in een digitale hield Tootill (Engeland) een interessante lezing.

De analoge grootheid wordt voorgesteld door de hoek waarover een as gedraaid is. Dit kan men digitaliseren door 3 ringen om de as aan te brengen, waarvan de eerste in 10, de tweede in 100, en de derde in 1000 segmenten is verdeeld, zodanig dat de eerste ring (d.m.v. een afleesborstel) de honderdtallen, de tweede de tientallen en de derde de eenheden afleest. Wanneer echter de honderdtallenborstel iets te vroeg het nieuwe cijfer aangeeft (en een exacte synchronisering der verschillende decimalen is technisch zeer moeilijk) dan ontstaan foutieve aflezingen, bij de overgang van 199 op 200 bv. 299 of 209.

Het is mogelijk om met een extra afleesborstel bovenstaande fout te vermijden, maar de gehele apparatuur wordt dan wel ingewikkeld. Daarom beveelt Tootill een andere weg aan. In de eerste plaats binair aflezen i.p.v. tientallig. Binair aflezen is nl. eenvoudiger dan tientallig, en bezit bovendien voordelen wanneer het resultaat door een binair werkende rekenmachine verder bewerkt moet worden. De in de voorgaande alinea vermelde fout wordt nu vermeden door de binaire code zodanig te permuteren dat opeenvolgende getallen steeds slechts in één bit verschillen. Dit blijkt dan nog zodanig te kunnen geschieden, dat de laatste $(n-1)$ bits van een getal gelijk zijn aan de eerste $(n-1)$ bits van het volgende getal (n stelt voor het totale aantal bits). Hiermee is de mogelijkheid geschapen om de n afleesringen door één afleesring te vervangen, die dan door n afleesborstels (of fotocellen) wordt afgetast.

Colloquium 11

Juncosa (U.S.A.) hield een voordracht over het oplossen van n simultane elliptische partiële differentiaalvergelijkingen in n onbekenden met 3 afhankelijk veranderlijken. Voor de I.B.M. gebruikers komt dit binnenkort beschikbaar als subroutine, die dus zonder meer in het programma van deze machines kan worden gebruikt. (Wellicht van belang voor de numerieke weersverwachting). Ook vergelijkingen, die parabolisch in de tijd zijn, kunnen hiermee opgelost worden, door een iteratie proces waarbij in elke stap een elliptische vergelijking wordt opgelost. Legras (Frankrijk) hield een lezing over het approximeren van een functie door polynomium of een rationale functie. Door een soort relaxatieproces worden de coëfficiënten in deze functie door de machine zelf bepaald.

Lotkin (U.S.A.) hield een verhandeling over het bepalen van de wortels van de karakteristieke vergelijking van een matrix. De methode is getest op

de I.B.M. 704 en gaf bevredigende resultaten.

Robertson (Engeland) gaf een methode voor het oplossen van gewone differentiaalvergelijkingen door een stap voor stap methode. De differentiaalvergelijking

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

kan men vervangen door een recurrente betrekking:

$$y_{n+1} = y_{n-1} + \frac{h}{3} (f_{n+1} + 4f_n + f_{n-1}) - T_n$$

waarin T_n , de restterm, van de orde h^5 is, en h de toename van x is per stap. Soms kan het echter voordeel hebben om een minder nauwkeurige benadering te nemen, waarbij men dan echter de beschikking krijgt over een parameter a :

$$y_{n+1} = ay_n + (1-a)y_{n-1} + \frac{h}{12} \left[(4+a)f_{n+1} + 8(2-a)f_n + (4-5a)f_{n-1} \right] + R_n$$

waarin nu R_n van de orde h^4 is, maar a nog nader gekozen kan worden. Van deze keuze kan men gebruik maken om de zgn. parasitaire oplossing (een probleem dat ook in de numerieke weersverwachting bekend is) te onderdrukken.