KM 22(1986)
pag 13 - 33

Een pseudo random number-generator voor 16-bits microcomputers

G.P. Cnossen *

Drs. J.N. Termeer **

Samenvatting

Generatoren die reeksen pseudo random getallen voortbrengen zijn reeds jarenlang in gebruik op wetenschappelijk, technisch en commercieel gebied. De theorievorming met betrekking tot dit soort generatoren is omvangrijk. Daarenboven zijn vele toetsen ontwikkeld die gehanteerd kunnen worden bij de beoordeling van de kwaliteit van de door de generatoren geproduceerde reeksen.

Bezien in het licht van de tendens om op de eerder genoemde gebieden steeds meer gebruik te gaan maken van microcomputers, is het interessant te beoordelen of dezelfde typen generatoren die voor mainframes gebruikt worden, in al dan niet gewijzigde vorm geschikt zijn voor implementatie op microcomputers. In dit artikel wordt beschreven hoe een multiplicatief-congruente generator werd geimplementeerd op een 16-bits micro-computer. Hierbij zullen de resultaten van een aantal toetsen gepresenteerd worden met behulp waarvan de 'kwaliteit' van de gegenereerde reeksen beoordeeld kan worden.

Dit artikel is in drie afzonderlijke hoofdstukken ingedeeld.

Het eerste hoofdstuk heeft een inleidend karakter. Het wordt geopend met een uitgebreidere motivering voor het uitvoeren van het onderzoek (paragraaf 1.1). In paragraaf 1.2 worden enkele kanttekeningen geplaatst bij de begrippen 'randomness', 'quasi-randomness', en 'pseudo-randomness'. In paragraaf 1.3 wordt vervolgens iets dieper ingegaan op random getallen-generatoren, m.n. multiplicatief-congruente. Een bespreking van een aantal mogelijke toetsen ter beoordeling van de kwaliteit van de gegenereerde reeksen volgt in paragraaf 1.4.

Hoofdstuk 2 bevat een beschrijving van de op een micro-computer geimplementeerde generator, en van de met deze generator behaalde testresultaten. De specificatie van de generator is opgenomen in paragraaf

* Student Economie aan de Rijksuniversiteit te Groningen
** Ten tijde van het schrijven van deze bijdrage, Wetenschappelijk hoofdassistent aan de Economische Faculteit van de Rijksuniversiteit te Groningen
Rijksuniversiteit Groningen, Postbus 800, 9700 AV Groningen
Tel. : (050-)633686

2.1. In paragraaf 2.2 wordt vervolgens aangegeven welke toetsen uit paragraaf 1.4 uitgevoerd werden. De resultaten die daaruit verkregen werden zijn opgenomen in paragraaf 2.3, waarna deze in paragraaf 2.4 kort besproken zullen worden.

In hoofdstuk 3 ten slotte wordt aandacht besteed aan de praktische toepasbaarheid van de generator, waarbij onder meer de cycluslengte van de gegenereerde reeksen aan de orde komt.

1.1 Motivering van het onderzoek.

Random getallen worden tegenwoordig voor zeer uiteenlopende doeleinden gehanteerd, onder meer in de numerieke analyse, in de statistiek (bij het trekken van steekproeven), en bij stochastische simulatie-modellen. Op al deze terreinen wordt meestal gebruik gemaakt van een computer als (onmisbaar) hulpmiddel.

Vele van de hedendaagse toepassingen, bijvoorbeeld op het gebied van de stochastische simulatie, zouden onmogelijk zijn zonder efficiente technieken om een reeks random getallen voort te brengen welke herhaalbaar is. Bij gebruik van random getallen die langs 'natuurlijke weg' (bijvoorbeeld m.b.v. een radiobuis) geproduceerd zijn kan herhaalbaarheid slechts gerealiseerd worden ten koste van een grote inefficientie. Om de gewenste combinatie van herhaalbaarheid en efficientie toch te realiseren zijn in de loop der jaren verschillende generatie-technieken ontwikkeld. Hierbij wordt de getallenreeks door een routine in het computerprogramma zelf (de generator) op gang gebracht en gehouden. Vele van deze generatoren produceren reeksen uniform verdeelde getallen op het interval <0,1>.

De aldus geproduceerde reeksen zijn in beginsel voorspelbaar, en kunnen derhalve strikt gesproken niet langer random genoemd worden. In dit verband spreekt men dan ook over pseudo- resp. quasi-random reeksen (zie ook paragraaf 1.2). De beoordeling van de mate van randomness geschiedt doorgaans met behulp van statistische toetsen, waarvan er eveneens verschillende zijn ontwikkeld. Er bestaat op dit punt een overweldigende hoeveelheid literatuur, met name uit de jaren vijftig, zestig en begin jaren zeventig. Halton [7] geeft een uitgebreid literatuuroverzicht van zowel generatoren als toetsen.

De recente ontwikkeling van generatoren en toetsen met steeds fraaiere theoretische eigenschappen heeft in de meeste gevallen geen belangrijke consequenties voor de praktische toepassingen gehad. Zo is een nog steeds veel gebruikte en doorgaans bevredigende techniek voor het genereren van de reeksen reeds aan het eind van de jaren '40 ontwikkeld (de lineair-congruente generator, zie Lehmer [15]). Deze techniek is in gebruik op hoofdzakelijk 'grote' computers (mainframes).

In de afgelopen jaren kon een penetratie geconstateerd worden van het gebruik van microcomputers op terreinen die tot voor kort aan 'mainframes' voorbehouden leken. De toepassingen van random getallen op de gebieden waarop aan het begin van dit artikel gedoeld werd lijken hierop echter een uitzondering te vormen. Bezwaren tegen het gebruik van stochastische technieken op microcomputers lijken zich op twee hoofdpunten te concentreren.

Allereerst wordt de opslagcapaciteit van microcomputers als een potentiele 'bottleneck' ervaren, aangezien bij de meeste toepassingen veel random getallen en (tussen-)resultaten verwerkt moeten worden. Echter, de tendens naar grotere interne, resp. virtuele geheugens voor professionele microcomputers is duidelijk aanwezig. Daarbij neemt de prijs/capaciteitsverhouding van microcomputers nog steeds af. Het lijkt er dus op dat van een structureel bezwaar geen sprake kan zijn.

Het tweede punt van kritiek heeft betrekking op de geschiktheid van microcomputerapparatuur en -programmatuur voor het genereren van random reeksen van voldoende lengte en 'kwaliteit'. In verschillende van de (ook oudere) theoretische publicaties wordt reeds gewezen op het gevaar dat specifieke problemen kunnen ontstaan indien computers met korte woordlengten gebruikt worden [1,3,13]. Microcomputers hebben over het algemeen een woordlengte van 16 bits of minder, hetgeen in vergelijking met woordlengten van mainframes gering is.

Het lijkt interessant te beoordelen of dezelfde typen generatoren die voor mainframes gebruikt worden al dan niet in gewijzigde vorm geschikt zijn voor implementatie op microcomputers. In het verslag van het onderzoek zal in hoofdstuk 2 aandacht besteed worden aan het kwaliteitsaspect, en in hoofdstuk 3 aan het aspect van de cycluslengte.

1.2 Randomness, quasi-randomness en pseudo-randomness.

In een reeks random getallen is elk getal uit de reeks onafhankelijk van alle eerdere (en volgende) getallen. Als nu met behulp van een programma-routine een reeks random getallen gegenereerd wordt, is dit niet meer het geval : de reeks is deterministisch bepaald. John von Neuman [18] zegt dan ook : 'Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.'. De gegenereerde getallen zijn strikt gesproken dan ook niet random te noemen. Indien ze desalniettemin een onderzoek m.b.t. randomness met goed gevolg doorstaan worden ze pseudo random getallen genoemd. Dergelijke getallenreeksen worden voor uiteenlopende doeleinden gebruikt, en het zijn meestal deze reeksen die geproduceerd worden door routines die in software-bibliotheken aanwezig zijn.

Die reeksen welke geen claim leggen op random gedrag maar wel een of meer eigenschappen bezitten waarin ze 'effectief random' [1]) zijn, worden quasi random reeksen genoemd. Deze reeksen kunnen in sommige gevallen worden gebruikt als alternatief voor pseudo random reeksen teneinde de efficientie van het rekenproces te vergroten.

Het is niet mogelijk om van een reeks getallen aan te tonen dat zij volledig random is. Het aantonen van het ontbreken van elk bestaanbaar onderliggend patroon is namelijk conceptueel en praktisch onmogelijk. In de praktijk wordt derhalve een beperkte verzameling toetsen gekozen met behulp waarvan de afwezigheid van bepaalde patronen beoordeeld wordt. Wil een random reeks praktisch toepasbaar zijn, dan moet deze in ieder geval de bedoelde toetsen doorstaan. Het blijft daarnaast echter altijd mogelijk dat de reeks desalniettemin voor sommige experimenten ongeschikt is. Een pseudo (of quasi) random reeks moet om deze reden idealiter vooraf getoetst worden op de aanwezigheid van specifieke patronen welke de reeks voor toepassing in deze experimenten minder geschikt maken.

Het toegepassen van toetsen op deterministisch gegenereerde reeksen geeft al met al niet meer dan een indicatie van de bruikbaarheid van de reeksen. Het beoordelen van de random reeksen met behulp van statistische toetsen blijft een subjectief gebeuren, al was het alleen maar door de keuze van de toetsen.

1.3 Random getallen-generatoren.

Omdat iedere cumulatieve distributiefunctie verkregen kan worden door een transformatie van uniform verdeelde random reeksen op het interval $\langle 0,1 \rangle$ [7], is het voldoende deze verdeling beschikbaar te hebben [2]). Een veel gebruikte en veelal bevredigende wijze om dergelijke uniform verdeelde reeksen te genereren is met behulp van een lineair-congruente generator. Dit type generator werd door D.H. Lehmer geintroduceerd [15]. De algemene vorm van een lineair-congruente generator is :

1. Bijvoorbeeld : de uniformiteit op een sub-interval is beter dan verwacht kan worden van een echte random reeks.
2. Dit neemt niet weg dat voor het genereren van aselecte trekkingen uit bepaalde verdelingen specifieke technieken bestaan, waarbij overigens ook gebruik gemaakt wordt van random getallen [12].

$X_i = a * X_{i-1} + c \pmod m$ met $i = 1,2,3\ldots,$

$a \geq 0,$

$c \geq 0,$

$m > a,$

$m > c,$

$m > X_0,$

a,c,m,X_0 geheeltallig.

De getallenreeks $U_i=X_i/m$ nemen we dan als de op $\langle 0,1\rangle$ uniform verdeelde reeks (of $[0,1\rangle$ indien $c \neq 0$).

Als $c = 0$ spreken we van een multiplicatief-congruente methode of 'power residue'-methode.

Als $c \neq 0$ spreken we van de 'mixed congruente' methode.

Het zal duidelijk zijn dat een op deze manier gegenereerde reeks X_i zichzelf na een bepaald aantal getallen gaat herhalen. Dit aantal wordt de periode van de generator genoemd. Er wordt op deze wijze een eindige verzameling ongelijke gehele getallen gevormd. Onder de gegeven omstandigheden resulteert de eerder genoemde afleiding van de reeks U_i uit de reeks X_i a forteriori in een eindige verzameling ongelijke rationele getallen.

De lengte van de periode hangt af van de waarden van de multiplier a, het increment c, de modulus m en de startwaarde X_0 (ook wel 'seed' genoemd).

Voor de multiplicatief-congruente generator (d.i. $c = 0$) is de maximale cycluslengte $m - 1$ indien we de modulus m kiezen als priemgetal terwijl X_0 oneven is en $a = 3$ of $5 \pmod 8$, ofwel $a = 8T+3$ of $a = 8T-3$ $(T = 0,1,2..)$. Nemen we m als een tweemacht, d.w.z. $m = 2^{\alpha}$ en α geheeltallig, dan is de maximale cycluslengte $m/4$, onder dezelfde voorwaarden voor a en X_0 als hierboven geformuleerd werden. Voor afleiding en bewijs hiervan zie [5,10]. Het voordeel van het gebruik van de laatste methode op een binaire computer is de mogelijkheid de binaire shiftoperatoren te benutten waardoor het generatieproces zeer snel kan verlopen. De maximale periode (d.i. $m/4$) wordt verkregen indien we voor α de woordlengte minus een nemen. Immers, het eerste bit wordt doorgaans gebruikt om het teken van een getal in het computerwoord vast te leggen. Bij deze keuze wordt echter het aantal elementen van de eerder genoemde eindige verzameling verder beperkt. Overigens blijkt dat de gegenereerde waarden van X_i regelmatig verspreid liggen in het interval $\langle 0..m\rangle$: als X_0 voldoet aan $X_0 = 1 \bmod 4$ voldoen

alle m/4 oneven waarden van de X_i's aan $X_i = 1 \mod 4$; analoog indien $X_0 = 3 \mod 4$ [3]).

In het verleden is het gedrag van de door lineair-congruente generatoren gegeneerde reeksen getallen (op mainframes) beoordeeld met behulp van een groot aantal verschillende toetsen [1,4,9,11,13,17]. De multiplicatief-congruente generator heeft daarbij over het algemeen tot iets bevredigender resultaten geleid dan de mixed generator [13,17], en is bovendien sneller [9,13,17].

1.4 Een aantal mogelijke toetsen.

De eersten die toetsen ter beoordeling van de kwaliteit van de gegenereerde reeksen rationele getallen U_i voorstelden waren Kendall en Babington-Smith [11]. Zij stelden de frequency-test, de serial-test, the gap-test en de poker-test voor. Later is deze lijst door verschillende auteurs uitgebreid. Voor uitgebreidere beschrijvingen van de toetsen wordt verwezen naar de literatuur terzake.

1. De frequency-test.

Het ligt voor de hand te verlangen dat de reeks getallen uniform verdeeld is op $\langle 0,1\rangle$ (resp. $[0,1\rangle$). Om deze eigenschap te toetsen verdelen we dit interval in k gelijke sub-intervallen. Vervolgens bepalen we de frequentie f_i, d.i. het aantal getallen in het i-de subinterval. Dan heeft voor een reeks van n getallen, bij benadering,

$$\text{CHISQ1} = k/n \sum_{i=1}^{k} (f_i - n/k)^2 \quad \text{een chi-kwadraat-verdeling,}$$

met k-1 graden van vrijheid [4]). We moeten er hierbij om denken dat de theoretische frequentie n/k groter dan vijf dient te zijn [4,5,10,11,13,17].

3. Vergelijk dit met hetgeen door Law en Kelton [14, pag. 225] gesteld wordt : "Furthermore, we generally shall not know where these m/4 integers will fall; i.e., there might be unacceptably large gaps in the X_i's obtained."
4. Het resultaat kan ook getoetst worden met de Kolmogorov-Smirnov-toets met $F(x) = x$, voor $0 < x < 1$.

2. De serial-test.

Een eenheidsvierkant wordt verdeeld in k gelijke cellen. Neem vervolgens het paar random getallen (U_i, U_{i+1}) als coordinaat van een punt in het eenheidsvierkant. Doe dit voor i = j..j+n-1, dus in totaal n-maal. Bepaal dan de frequenties f_{ij} van iedere cel (i,j). Bereken vervolgens

$$STAT = k^2/n \sum_{i,j=1}^{k} (f_{ij} - n/k^2)^2$$

met als voorwaarde dat $n/k^2 > 5$. Omdat de waarnemingen niet onafhankelijk zijn [5], kunnen we er niet van uitgaan dat STAT een chi-kwadraat-verdeling heeft met k^2-1 graden van vrijheid. Good [6] heeft laten zien dat (STAT - CHISQ1) asymptotisch een chi-kwadraat-verdeling heeft met $(k^2 - k)$ graden van vrijheid en dat (STAT - 2 * CHISQ1) asymptotisch chi-kwadraat verdeeld is met $(k-1)^2$ graden van vrijheid. Deze toets kan ook worden uitgevoerd door de paren (U_{2i}, U_{2i+1}) te nemen. Dan is er geen afhankelijkheid meer in de waarnemingen en kunnen we STAT toetsen als chi-kwadraat-verdeling met (k^2-1) graden van vrijheid. Het aantal dimensies kan worden uitgebreid tot drie (triplets) of meer [4,11]. Ook kan deze toets worden uitgevoerd door de paren (U_{3i}, U_{3i+i}) te nemen. We passen dan de serial toets toe met lag 2.

3. De gap-test.

In deze toets bepalen we de lengte van het 'gap' dat zit tussen het optreden van twee getallen die beide in een sub-interval voorkomen. Als a en b twee getallen zijn met $0 < a < b < 1$, dan kijken we naar de lengte van de subreeksen $U_i, U_{i+1}, \ldots U_{i+k}$ waarbij U_{i+k} in het interval [a,b] ligt en $U_i, U_{i+1}, \ldots U_{i+k-1}$ niet. Dit is dan een gap met lengte k.

Om de toets uit te voeren nemen we een reeks van n getallen en bepalen we het aantal malen dat een gap van lengte k voorkomt. Dit noemen we f_k met $0 \leq k < t$. Verder bepalen we het aantal malen dat een gap voorkomt met $k \geq t$, f_t. Dit om er voor te zorgen dat de theoretische frequenties f_i (i = 0..t) alle groter dan vijf zijn.

De waarschijnlijkheid dat een gap voorkomt met lengte 0 is $p_0 = p$ met p = b-a. De waarschijnlijkheid van een gap met lengte 1 is $p_1 = (1-p)p$, de waarschijnlijkheid van een gap met lengte 2 is $p_2 = (1-p)^2 p$, enz. De waarschijnlijkheid van een gap met lengte t of groter is $p_t = (1-p)^t$.

5. Immers, de y-coordinaat van een punt is de x-coordinaat van het daaropvolgende punt.

De toetsingsgrootheid

$$CHISQ2 = \sum_{k=0}^{t} (f_k - N * p_k)^2/(N*p_k)$$

is dan chi-kwadraat verdeeld met t graden van vrijheid [11,13].
Nemen we a = 0 en b = 1/2 (de mediaan van de te toetsen reeks getallen) dan is de gap-toets identiek aan een andere toets, n.l. die op 'runs above and below the median' [1,4,10,13].

4. De poker-test.

Verdeel de getallen in groepen van vijf en tel het aantal malen dat een bepaalde groepenrij voorkomt, b.v. full-house, three of a kind enz. De waarschijnlijkheden van het optreden van een bepaalde groepering zijn berekend door Herrman [8].

5. Maximum of n.

De functie $W = \max(U_i, U_{i+1}..U_{i+n})$ heeft theoretisch de distributiefunctie $P(W < a) = F(a) = a^n$ voor $0 < a < 1$, en dus zou $F(W) = [\max(U_i,..U_{i+n})]^n$ uniform verdeeld moeten zijn op $\langle 0,1 \rangle$ [9,10,13,17]. Deze toets kan bijvoorbeeld toegepast worden met n = 2,3,4,... . Analoog kan deze toets ook worden uitgevoerd voor het minimum van een reeks van n getallen.

6. Serial correlation.

De serial correlation met lag k wordt gedefinieerd als

$$r_k = Cov(U_i, U_{i+k}) / \sqrt{(Var(U_i) * Var(U_{i+k}))} \quad \text{met } k = 1,2,3.. .$$

Voor een random reeks zouden alle correlaties 0 moeten zijn. Maar omdat (U_0, U_1) niet geheel onafhankelijk is van (U_1, U_2) zullen de correlaties niet precies nul zijn. Verwacht wordt dat r_k in 95% van de gevallen tussen de grenzen a - 2b en a + 2b ligt met

$$a = -1/(n+1) \text{ en } b = (1/(n-1)) * \sqrt{(n(n-3)/(n+1))} \quad \text{met } n > 2. \text{ [13]}$$

Voor de meeste problemen zullen de correlaties voor de lags van lagere orde van meer belang zijn dan die van de hogere-orde lags.

7. Run-test.

a. Oorspronkelijke vorm

Voor een reeks van n getallen uit een random reeks definieren we een (n-1)-bitreeks waarvan de i^e term gelijk aan 0 is als $U_i < U_{i+1}$, en gelijk aan 1 als $U_i > U_{i+1}$. Noem k de lengte van een reeks nullen omsloten door enen of een reeks enen omsloten door nullen. De verwachte waarden zijn [8] :

$(2n-1)/3$ voor het totale aantal runs

$(1/12) * (5n + 1)$ voor het aantal runs met lengte 1

$(1/60) * (11n - 14)$ voor het aantal runs met lengte 2

$2\,[(k^2 + 3k + 1)\,n - (k^3 + 3k^2 - k - 4)] \,/\, (k+3)!$
voor het aantal runs met lengte k $\qquad (k < n-1)$.

Het verwachte aantal runs met lengte k of groter bedraagt

$[2(k + 1)n - (k^2 + k-1)] \,/\, (k+2)!$

De aantallen waarnemingen van runs van elke onderscheiden lengte werden vergeleken met de respectieve verwachte waarden, en de chi-kwadraat-toetsingsgrootheid werd bepaald [4]. Maar omdat de opeenvolgende runlengtes niet onafhankelijk zijn is dit niet terecht. Een andere vorm van de run-test is dan ook voorgesteld door H. Levene en J. Wolfowitz in 1944 [16]. Hierin wordt onderscheid gemaakt tussen runs up en runs down.

b. Runs up en runs down

Voor een reeks van n getallen definieren we een n-bitreeks. Het eerste bit wordt een 1. De i-de term van de reeks is gelijk aan de (i-1)-de term als $U_i > U_{i-1}$ en verschilt hiervan als $U_i < U_{i-1}$. Het aantal runs up van lengte k wordt vervolgens bepaald door alle reeksen enen en nullen van lengte k te tellen. Bereken vervolgens :

$$\text{CHISQ3} = 1/n \sum_{i,j=1}^{6} ((\text{RUNLENGTE}[i] - nb_i)\,(\text{RUNLENGTE}[j] - nb_j)\,a_{ij})$$

(hierin bevat RUNLENGTE[6] het aantal runs up met lengte 6 of groter), met a_{ij} de elementen van matrix A, [13] :

$$A = \begin{bmatrix} 4529.4 & 9044.9 & 13568 & 18091 & 22615 & 27892 \\ 9044.9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 13568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{bmatrix}$$

en b_i de elementen van matrix B :

$$B = [1/6 \quad 5/24 \quad 11/120 \quad 19/720 \quad 29/5040 \quad 1/840]$$

CHISQ3 heeft een chi-kwadraat-verdeling met 6 graden van vrijheid voor grote n [13,16]. Op dezelfde wijze kan deze toets uitgevoerd worden met runs down.

8. De spectral-test.

Deze toets werd in 1965 voorgesteld door R.R. Coveyou en R.D. Macpherson [3]. De wiskundige achtergrond van deze toets is gelegen in de eindige Fourier-transformatie van een functie, gedefinieerd op een eindig bereik. In tegenstelling tot de andere toetsen, die toegepast worden op de gegenereerde reeksen wordt in deze toets de multiplier (de a-waarde) getoetst. Hij kan dus uitstekend toegepast worden als een apriori toets, d.w.z. voordat er ook maar een getal gegenereerd is, met het doel goede multipliers op te sporen. D.E. Knuth [13] zegt van deze toets dat alle lineair-congruente random generatoren waarvan wij nu weten dat ze geen voldoende random reeksen voortbrengen, falen op deze toets. Voor de theorie achter deze toets zie [3,13].

In de praktijk werkt deze toets als volgt :

Bepaal uit de waarde van de multiplier de zogeheten minimale wavenumbers $v_1, v_2, .., v_n$. Het wavenumber v_i geeft een indicatie van het aantal bits j van elk getal dat als onafhankelijk over de i opeenvolgende getallen beschouwd kan worden. Bijvoorbeeld : als $v_3 = 256 \ (= 2^8)$ kunnen, wat betreft de onafhankelijkheid van opeenvolgende triplets, in binaire notatie alleen de eerste 8 bits als random beschouwd worden. De waarde v_n wordt als volgt bepaald :

minimaliseer $v_n = \sqrt{(s_1^2 + s_2^2 + .. + s_n^2)}$ onder de voorwaarde dat

$$s_1 + as_2 + a^2s_3 + .. + a^{n-1}s_n == 0 \ (\text{mod } P)$$

waarbij a de waarde van de multiplier is en $s_1, s_2, .. s_n$ gehele getallen, niet alle gelijk nul. P is gelijk aan de modulus m van de lineair-congruente generator als deze mixed is, gelijk aan m/4 voor een

multiplicatief-congruente generator met a = 8T + 5, en voor dezelfde generator gelijk aan m/8 indien a = 8T + 3.

Voor een multiplicatief-congruente generator met een modulus $m = 2^{15}$ en een multiplier die voldoet aan a = 8T + 3 geldt de volgende correspondentie tussen het wavenumber v_n en j :

wavenumber	j
v_2 <= 97.3	6
v_3 <= 22.6	4
v_4 <= 11.3	3
v_5 <= 7.5	3

Knuth [13] stelt voor om te bepalen :

$$C_n = \pi \left({}^{n/2} v_n^n \right) / ((n/2)!P)$$

ofwel $C_1 = 2v_1 \quad / (m/4)$ is altijd 2

$C_2 = \pi v_2^2 \quad / (m/4)$

$C_3 = (4/3)\ \pi v_3^3 \quad / (m/4)$

$C_4 = (1/2)\ \pi^2 v_4^4 \quad / (m/4)$

en te stellen dat een generator de toets doorstaat als C_2, C_3 en $C_4 \geq 0.1$, en er 'met vlag en wimpel' doorkomt als C_2, C_3 en $C_4 \geq 1$. Hierbij dienen de waarden v_n ook afzonderlijk in beschouwing genomen te worden, want het voldoen aan deze toets garandeert voor lineair-congruente generatoren met kleine moduli nog niet een voldoende mate van randomness [13].

2. Specificatie en testresultaten van de op een micro-computer geimplementeerde generator.

2.1 De generator.

Voor het onderzoek werd gebruik gemaakt van een multiplicatief-congruente generator, geimplementeerd op een 16-bit microcomputer [6]). Een bit is bestemd voor het teken van een getal. Als moduluswaarde werd dan ook genomen : 2^{15} (= 32768). Dit houdt in dat de periode 2^{13} (= 8192) is.

Vervolgens werd m.b.v. de spectral-test [7]) een aantal multiplier-waarden bepaald. Drie op grond van het criterium dat C_2, C_3 en C_4 zo hoog zouden moeten zijn dat de resulterende verzameling mogelijke multipliers klein zou zijn. Uit de daarbij verkregen 8 waarden werden resp. een middelste en de uiterste waarden van de multipliers gekozen. Dit zijn de multipliers 5133, 15045, 31429. De drie anderen zijn 53, 173, 32565. Zij kwamen uit de groep getallen waarvoor C_2, C_3 en C_4 slechts groter dan 1 hoefden te zijn. Met opzet zijn deze waarden gekozen omdat vaak gesteld wordt dat grote en kleine waarden voor a vermeden moeten worden [3,13]. 173 ligt dicht in de buurt van $\sqrt{m}$ = 181; volgens Coveyou minimaliseert dit de correlatie tussen twee opeenvolgende getallen [2]. Een overzicht van de bij deze multipliers behorende v- en C-waarden is hieronder afgedrukt. Aangezien het praktisch belang van de waarde v_n afneemt naarmate n hoger is zijn alleen de waarden v_2, v_3 en v_4 bepaald.

multiplier	v_2	v_3	v_4	C_2	C_3	C_4
53	53.0	12.9	6.5	1.07	1.10	1.08
173	77.0	15.9	7.9	2.27	2.06	2.35
5133	89.6	19.0	9.3	3.08	3.51	4.51
15045	89.6	19.0	9.3	3.08	3.51	4.51
31429	89.6	19.0	9.3	3.08	3.51	4.51
32565	82.4	16.3	7.9	2.60	2.21	2.35

6. De gebruikte apparatuur bestond uit IBM en IBM-compatibele microcomputers. De programmatuur werd geschreven in TURBO-Pascal, in **single** precision, d.w.z. dat slechts minimale eisen aan de mogelijkheden tot getalrepresentatie gesteld werden. Daar iedere programmeertaal op microcomputers met een 16-bits-architectuur deze minimale mogelijkheid bezit kunnen de verkregen resultaten als generiek voor alle soortgelijke apparatuur en programmatuur beschouwd worden.
7. Deze en alle andere toetsen werden uitgevoerd op een Cyber 170/760 mainframe.

2.2 De toegepaste toetsen.

Op de bij de multipliers horende gegenereerde reeksen (in het navolgende aangeduid met 'RAx', waarbij x de desbetreffende multiplier is) hebben wij vervolgens de onderstaande toetsen toegepast :

- De frequency-test : hoewel geen sterk discriminerende toets, toch toegepast vanwege eenvoud en vergelijkbaarheid met andere onderzoeken.
- De serial-test met lag 1 t/m 6 en met triplets, quadruples en quintuples. (Het toepassen van deze toets met meer dimensies is een nauwkeuriger toets op hetzelfde als de poker-test of maximum of n-test [13]).
- De run-test, door Downham en Roberts [4] een zeer goede toets genoemd : 'if a generator failed any other test, then it also failed this test.' In tegenstelling tot hen hebben wij de run-test toegepast op de wijze van Levene en Wolfowitz, d.w.z. met onderscheid in runs up en runs down.

Alle toetsen behalve de run-test zijn driemaal uitgevoerd (d.w.z. 3 runs); elke run start op een andere plaats in de cyclus.

Tevens werd een toets op seriele correlatie (voor lag 1 t/m 100) uitgevoerd. Een overzicht van de waarden van de correlaties zou teveel ruimte in beslag nemen. Bij alle waarden van de multiplier a die door ons gebruikt zijn, vielen de berekende correlaties binnen de grenzen zoals gesteld in paragraaf 1.4.

De genoemde toetsen werden uitgevoerd op twee onderscheiden groepen met, in beginsel, 200 resp. 1000 waarnemingen. Deze groepen worden in het vervolg aangeduid met resp. 'groep I' en 'groep II'.

- Groep I.
  De frequency-test is uitgevoerd met 200 getallen en 20 subintervallen, d.i. k = 20.
  De serial-test voor lag 1 t/m lag 6 is uitgevoerd met 200 paren, dus voor b.v. lag 3 zijn 800 getallen nodig voor het vermijden van afhankelijkheid in de waarnemingen. Voor triplets zijn 27 cellen gebruikt en 600 getallen die 200 waarnemingen vormen. Voor quadruples zijn 16 cellen en 800 getallen gebruikt die 200 waarnemingen vormen, en voor quintuples 32 cellen en 1000 getallen die eveneens 200 waarnemingen opleveren.
  De run-test is niet uitgevoerd daar n (zie paragraaf 1.4) niet groot genoeg is om een chi-kwadraat-verdeling op voldoend nauwkeurige wijze te benaderen.

- Groep II.
  De frequency-test is uitgevoerd met 1000 getallen en 100 subintervallen, d.i. k = 100.
  De serial-test voor lag 1 t/m lag 6 is uitgevoerd met 1000 paren. Voor triplets zijn 125 cellen gebruikt en 3000 getallen die 1000 waarnemingen vormen. Voor quadruples zijn 81 cellen en 4000 getallen gebruikt (1000

waarnemingen), en voor quintuples 32 cellen en 5000 getallen (1000 waarnemingen) [8]).
De run-test is tweemaal uitgevoerd met 4000 getallen. Dit om ervoor te zorgen dat n voldoende groot is om de chi-kwadraat-verdeling nauwkeurig genoeg te kunnen benaderen.

2.3 Testresultaten.

Het aantal waarnemingen is voldoende groot om bij de beoordeling van de testresultaten gebruik te maken van de chi-kwadraat-toets. De resultaten van de toetsingsprocedures staan vermeld in de tabellen.

Hierin staat bovenaan de waarde van de multiplier (RAx); in de linker kolom staat de naam van de toets, en, per rij, het volgnummer van de run. In de cellen staat de waarschijnlijkheid van het optreden van de geobserveerde chi-kwadraat-waarde, of een grotere waarde, onder de nulhypothese dat de reeks 'echt' random zou zijn. Een 'e' in een cel betekent dat deze waarschijnlijkheid kleiner dan of gelijk aan 0.01 is.

8. Voor bijvoorbeeld de toetsen met triplets werden drie groepen van 3000 getallen gebruikt; aangezien het totaal aantal beschikbare getallen 8192 bedroeg zouden sommige waarden X_i tweemaal gebruikt worden. Door te zorgen dat de beginwaarde van iedere run niet samenviel met enige eerste coordinaat van enig triplet gebruikt in de andere twee runs, werd voorkomen dat enig punt tweemaal gebruikt werd.

Tabel 1
Groep I : telkens 200 waarnemingen

		RA53	RA173	RA5133	RA15045	RA31429	RA32565
frequency	1e	0.37	0.07	0.39	0.08	0.26	0.36
	2e	0.84	0.42	0.31	0.95	0.55	0.29
	3e	0.42	0.93	0.41	0.88	0.60	0.88
serial, lag 1	1e	0.49	0.77	0.34	0.75	0.87	0.55
	2e	0.26	0.77	0.64	0.84	0.70	0.84
	3e	0.34	0.67	0.07	0.55	0.24	0.18
lag 2	1e	0.51	0.86	0.40	0.41	0.77	0.80
	2e	0.59	0.45	0.41	0.88	0.36	0.08
	3e	0.37	0.98	0.75	0.06	0.34	0.79
lag 3	1e	0.28	0.27	0.75	0.20	0.10	0.17
	2e	0.32	0.49	0.70	0.43	0.26	0.36
	3e	0.55	0.88	0.49	0.05	0.03	0.46
lag 4	1e	0.61	0.34	0.71	0.11	0.15	0.10
	2e	0.68	0.65	0.80	0.61	0.68	0.61
	3e	0.45	0.28	0.19	0.24	0.97	0.68
lag 5	1e	0.09	0.63	0.39	0.58	0.38	0.20
	2e	0.40	0.61	0.42	0.81	0.25	0.27
	3e	0.26	0.29	0.30	0.32	0.26	0.12
lag 6	1e	0.51	0.68	0.99	0.64	0.42	0.27
	2e	0.38	0.10	0.85	e	0.08	0.67
	3e	0.87	0.59	0.17	0.46	0.97	0.82
triplets	1e	0.27	0.59	0.43	0.79	0.20	0.15
	2e	0.48	0.98	0.30	0.05	0.88	0.80
	3e	0.23	0.70	0.90	0.55	0.67	0.49
quadruples	1e	0.04	0.27	0.72	0.33	0.33	0.07
	2e	0.15	0.48	0.73	0.58	0.63	0.64
	3e	0.98	0.30	0.96	0.18	0.21	0.99
quintuples	1e	0.56	0.08	0.26	0.96	0.10	0.74
	2e	0.29	0.17	0.24	0.96	0.25	0.07
	3e	0.89	0.68	0.92	0.77	0.33	0.59

Tabel 2
Groep II : telkens 1000 waarnemingen

		RA53	RA173	RA5133	RA15045	RA31429	RA32565
frequency	1e	0.89	0.90	0.87	0.98	0.49	0.74
	2e	0.89	0.63	0.87	0.92	0.51	0.99
	3e	0.70	0.56	0.58	0.79	0.22	0.90
serial, lag 1	1e	0.84	0.96	0.96	0.99	0.99	0.83
	2e	0.51	0.75	0.94	0.99	0.99	0.80
	3e	0.92	0.87	0.97	0.98	0.98	0.96
lag 2	1e	0.66	0.86	0.99	0.99	0.99	0.67
	2e	0.71	0.60	0.60	0.85	0.76	0.98
	3e	0.92	0.77	0.82	0.99	0.50	0.97
lag 3	1e	0.99	0.99	0.99	0.99	0.99	0.99
	2e	0.99	0.99	0.99	0.99	0.99	0.99
	3e	0.99	0.99	0.99	0.99	0.99	0.99
lag 4	1e	0.83	0.85	0.75	0.69	0.11	0.85
	2e	0.99	0.35	0.99	0.67	0.85	0.97
	3e	0.99	0.35	0.83	0.75	0.73	0.99
lag 5	1e	0.73	0.97	0.99	0.99	0.99	0.95
	2e	0.88	0.99	0.99	0.99	0.99	0.86
	3e	0.88	0.99	0.99	0.99	0.99	0.89
lag 6	1e	0.51	0.92	0.89	0.46	0.87	0.87
	2e	0.36	0.80	0.74	0.43	0.38	0.85
	3e	0.62	0.73	0.78	0.54	0.84	0.93
triplets	1e	0.72	0.95	0.99	0.91	0.85	0.55
	2e	0.83	0.26	0.77	0.98	0.97	0.69
	3e	0.86	0.11	0.57	0.87	0.90	0.88
quadruples	1e	0.98	0.99	0.99	0.99	0.85	0.54
	2e	0.99	0.99	0.98	0.99	0.97	0.77
	3e	0.99	0.99	0.96	0.98	0.99	0.42
quintuples	1e	0.50	0.56	0.51	0.19	0.61	0.96
	2e	0.68	0.74	0.47	0.28	0.16	0.23
	3e	0.94	0.59	0.14	0.51	0.93	0.93
runs up	1e	0.34	0.91	0.49	0.47	0.66	0.75
	2e	0.41	e	0.18	0.74	0.81	0.61
down	1e	0.04	e	0.79	0.33	0.86	0.55
	2e	0.15	0.37	0.96	0.21	e	0.10

2.4 Bespreking van de resultaten.

Aan het slot van paragraaf 1.2 werd reeds opgemerkt dat het beoordelen van random reeksen met behulp van statistische toetsen een subjectief gebeuren is. Onze navolgende interpretaties van de testresultaten moeten dan ook in dit licht gezien worden. In een poging de intersubjectiviteit op dit punt te vergroten zal allereerst aansluiting gezocht worden bij de literatuur.

MacLaren en Marsaglia [17] zeggen dat de waarden in de tabel voor elke generator [9]) zich zouden moeten gedragen als een verzameling getallen die uniform gekozen zijn op het eenheidsinterval. Er kunnen kleine of grote waarden zijn, maar een oververtegenwoordiging van zulke waarden is verdacht.

Voor de waarden vermeld in de eerste tabel is een dergelijke oververtegenwoordiging afwezig [vgl. 4,17]. Het gaat hierbij om relatief kleine deelverzamelingen van de gegenereerde getallen (sub-intervallen) waarover getoetst wordt. Uit de eerste tabel kan worden geconcludeerd dat een generator op dezelfde toets tot verschillende resultaten kan leiden (d.w.z. zowel hoge als lage waarden), afhankelijk van de uitgevoerde run. Vergelijkbare, niet afgedrukte resultaten van toetsen op sub-intervallen van dezelfde lengte, die op andere plaatsen in de cyclus starten, laten ditzelfde effect zien. Hieruit, en uit een vergelijking met tabel 2, kunnen twee conclusies getrokken worden. Ten eerste, dat enige voorzichtigheid in acht genomen moet worden bij de beoordeling van de mate van randomness van afzonderlijke generatoren op basis van een relatief klein aantal waarnemingen zoals in tabel 1. Ten tweede dat de variabiliteit van de resultaten op langere reeksen afneemt, zoals blijkt uit tabel 2. Met tabel 2 vergelijkbare, niet afgedrukte resultaten van sub-intervallen van dezelfde lengte, die op andere plaatsen in de cyclus starten, laten een soortgelijke stabiliteit zien.

Voor de run-test in tabel 2 geldt een iets andere interpretatie. Hier immers kan geen tendens verwacht worden waardoor een oververtegenwoordiging van (bijvoorbeeld) lange runs op een bepaald sub-interval gecompenseerd wordt wordt door een groot aantal korte runs op een ander sub-interval.

Op deze wijze beschouwd zijn de testresultaten die met de grotere sub-intervallen (tweede tabel) verkregen werden, verklaarbaar : hier komen relatief veel hoge waarden voor de frequency- en serial-test voor [10]). Het aantal getallen dat gebruikt werd om tot de resultaten in de tweede tabel te komen, is relatief groot (in sommige gevallen meer dan 50% van het totaal aantal getallen dat beschikbaar is).

9. In het voorgaande is deze term steeds gehanteerd om generatietechnieken (zoals de lineair-congruente) te kunnen benoemen. Wij zullen, uitsluitend in deze en de volgende paragraaf, de term 'generator' op een andere wijze hanteren, namelijk kortheidshalve ter aanduiding van elke afzonderlijke reeks RAx.
10. Een soortgelijk gedrag kan worden waargenomen bij generatoren met grotere moduli (typisch op mainframes), met dien verstande dat het optreden van relatief veel hoge waarden pas bij grotere aantallen waarnemingen begint.

Met betrekking tot de respectieve onderzochte generatoren kan samenvattend het volgende opgemerkt worden :

- RA173 is een generator die twijfels omtrent zijn gedrag oproept. Zie de run-test in de tweede tabel.
- Hetzelfde geldt, in iets mindere mate, voor RA31429.
- Onze eerdere opmerkingen over de waarde van de resultaten uit tabel 1 in het achterhoofd houdend, moet geen overdreven belang gehecht worden aan de extreem lage waarde van RA15045 voor de serial-test, lag 6, uit tabel 1.
- Tegen het gebruik van de generatoren RA53, RA5133 en RA32565 kunnen op grond van de verkregen resultaten geen overwegende bezwaren aangevoerd worden. Opvallend is het gedrag van de generator met multiplier 53, RA53. Er is geen verschil in gedrag t.o.v. de andere geconstateerd, terwijl vaak gewaarschuwd wordt voor (te) kleine multipliers [3,10,13].

Bij dit alles moet bedacht worden dat de multipliers van de geteste generatoren aan vooraf gestelde criteria dienden te voldoen om verkozen te worden (spectral-test). De met deze geselecteerde generatoren verkregen resultaten zijn zeker niet representatief voor generatoren met andere, willekeurig gekozen multipliers. Enige experimentatie op dit gebied toont dit al snel aan.

3. Praktische toepasbaarheid; cycluslengte.

Uit de testresultaten komt naar voren dat het ook met een micro-computer mogelijk is reeksen van voldoende kwaliteit te genereren. Uit een vergelijking van de tabellen in paragraaf 2.3 blijkt echter eveneens dat reeds bij relatief kleine aantallen waarnemingen hoge waarden verkregen worden voor de in de cellen afgedrukte waarschijnlijkheden. Dat wil zeggen dat, ook bij een gering aantal waarnemingen [11]), over het algemeen geen sprake is van een belangrijke over- of ondervertegenwoordiging van bepaalde getallen in zo'n sub-reeks. Een dergelijke conclusie kan van belang zijn voor het oordeel met betrekking tot de praktische toepasbaarheid van de reeksen voor bepaalde doeleinden.

Zo kan het snel bereiken van een hoge mate van uniformiteit een voordeel zijn, bijvoorbeeld bij een eerste beoordeling van het gedrag van de stochastische variabelen in een simulatiemodel; nabootsing van de algemene karakteristieken van de gewenste cumulatieve distributiefuncties (zie paragraaf 1.3) vindt relatief snel plaats. Dit voordeel kan overigens voor studies waarin meer detail gewenst is in een nadeel omslaan, mede omdat reeksen die met deze generatoren geproduceerd worden een tamelijk korte cycluslengte hebben.

11. Zie in dit verband ook voetnoot 10 in paragraaf 2.4.

In dit laatste kenmerk schuilt ook een direct ander gevaar. Een niet goed doordachte keuze van een startwaarde kan leiden tot ongewenste effecten : indien de startwaarde van enige replicatie overeenkomt met een eerder door de generator geproduceerde waarde, wordt de reeds eerder geproduceerde reeks vanaf dat punt in de cyclus herhaald, met alle nadelige gevolgen voor de (simulatie-)resultaten van dien. Indien de keuze niet goed doordacht of zelfs geheel arbitrair gedaan wordt, is de a priori-waarschijnlijkheid van het optreden van dergelijke ongewenste effecten groter dan bij gebruik van reeksen met grotere cycluslengten.

De beperkte cycluslengte van een op een microcomputer gegenereerde reeks kan dus een probleem vormen. We zullen daarom twee mogelijkheden ter vergroting ervan noemen.

Allereerst ligt het voor de hand andere typen generatoren dan de multiplicatief-congruente uit te proberen. Dit kunnen andere, op mainframes gangbare generatoren zijn, zoals de mixed congruente.

Een andere, veelbelovende aanpak die we willen noemen is het gebruikmaken van de zogeheten technieken van 'shuffling' [17]. Hierbij worden twee of meer generatoren, met eventueel zeer kleine cycluslengten, gecombineerd tot een samengestelde generator met een (zeer) grote cycluslengte. Verdere voordelen zijn, dat men gebruik kan blijven maken van goed onderzochte generatoren, zoals multiplicatief-congruente, en dat ook kwalitatief goede reeksen verkregen kunnen worden [17].

Wij achten verder onderzoek naar 'shuffling'-generatoren dan ook van groot praktisch belang. We willen dit artikel derhalve afsluiten met een suggestie die voortkomt uit onze ervaring met de implementatie van multiplicatief-congruente generatoren op microcomputers, en die wellicht relevantie kan hebben voor het ontwikkelen van nieuwe generatoren van het 'shuffling'-type.

De startwaarde X_0 van een multiplicatief-congruente generator moet, om een maximale periode te genereren, een oneven getal zijn. Alle gegenereerde getallen, dus voor deling door de modulus, zijn dan ook oneven. Het blijkt dat als voor de startwaarde geldt dat $X_0 = 4T + 3$, $T = 0,1,2..$, (d.w.z. $X_0 = 3 \bmod 4$) dit ook geldt voor alle gegenereerde $X_i$waarden. Hetzelfde geldt als de startwaarde voldoet aan $X_0 = 4T + 1$, $T = 0,1,2..$ (d.w.z. $X_0 = 1 \bmod 4$).

Als men twee multiplicatief-congruente generatoren combineert in een generator van het 'shuffling'-type, kan men dan ook het beste werken met twee startwaarden, waarvan de een voldoet aan $X_0 = 1 \bmod 4$ en de ander aan $X_0 = 3 \bmod 4$, zodat alle $2^{\alpha-2}$ oneven getallen gebruikt worden.

Literatuur

1. J.L. Allard, A.R. Dobell, T.E. Hull, Mixed congruential random number generators for decimal machines.
Journal of the Association for Computing Machinery, Vol. 10, no. 1, 1963, pp. 131-142.

2. R.R. Coveyou, Serial correlation in the generation of pseudo-random numbers.
Journal of the Association of Computing Machinery, Vol. 7, 1960, pp. 72-74.

3. R.R. Coveyou, R.D. Macpherson, Fourier analysis of uniform random number generators.
Journal of the Association for Computing Machinery, Vol. 14, no. 1, 1967, pp. 110-119.

4. D.Y. Downham, F.D.K. Roberts, Multiplicative congruential pseudo random number generators.
The Computer Journal, Vol. 10, no. 1, 1967, pp. 74-77.

5. A.T. Fuller, The period of pseudo random numbers generated by Lehmer's congruential method.
The Computer Journal, Vol. 19, no. 2, 1976, pp. 173-177.

6. I.J. Good, The serial test for sampling numbers and other tests for randomness.
Proc. Camb Phil soc., Vol. 49, 1953, p. 276.

7. J. H. Halton, A retrospective and prospective survey of the Monte Carlo method.
Siam Review, Vol. 12, 1970, pp. 1-63.

8. R.G. Hermann, The statistical evaluation of random number generating sequences for digital computers.
Office of Technical Services, US Dept. of Commerce, Apex-635, Washington DC, 1961.

9. T.E. Hull, A.R. Dobell, Mixed congruential random number generators for binary machines.
Journal of the Association for Computing Machinery, Vol. 11, no. 1, 1964, pp. 31-40.

10. T.E. Hull, A.R. Dobell, Random number generators.
Siam Review, Vol. 4, no. 3, 1962, pp. 230-254.

11. M.G. Kendall, B. Babington-Smith, Randomness and random sampling numbers.
Journal of the Royal Statistical Society, Vol. 101, 1938, pp. 147-166.

12. W.J. Kennedy,Jr, J.E. Gentle, Statistical computing
Marcel Dekker. Inc., 1980.

13. D.E. Knuth, The art of computer programming, Vol. 2, Seminumerical Algorithms.
Addison-Wesley Publishing Company Inc., 1969.

14. A.M. Law, W.D. Kelton, Simulation modeling and analysis.
McGraw-Hill Book Company, 1982.

15. D.H. Lehmer, Mathematical methods in large scale computing units.
Annals of the Computation Laboratory of Harvard University, No. 26,
Proceedings of a second Symposion on large-scale digital calculating machinery, 1951, p. 141.

16. H. Levene, J. Wolfowitz,
Annals of Mathematical Statistics, Vol. 15, 1944, pp. 163-165.

17. M.D. MacLaren, G. Marsaglia, Uniform random number generators.
Journal of the Association for Computing Machinery, Vol. 12, no. 1, 1965, pp. 83-89.

18. J. von Neumann, Various techniques used in connection with random digits.
Monte Carlo methods, National Bureau of Standards, Applied Mathematics Series no. 12, US Government Printing Office, Washington DC, 1951, pp. 36-38.