

**KERNFORSCHUNGSZENTRUM  
KARLSRUHE**

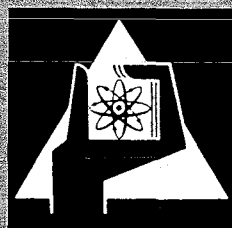
Januar 1971

KFK 1361

Institut für Neutronenphysik und Reaktortechnik  
Projekt Schneller Brüter

Strategien zur Beschleunigung zweidimensionaler SN-Verfahren  
mit Hilfe der Grobgittertechnik

W. Kinnebrock



GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.

KARLSRUHE



KERNFORSCHUNGSZENTRUM KARLSRUHE

Januar 1971

KFK 1361

Institut für Neutronenphysik und Reaktortechnik  
Projekt Schneller Brüter

Strategien zur Beschleunigung zweidimensionaler SN-Verfahren  
mit Hilfe der Grobgittertechnik

W. Kinnebrock

Gesellschaft für Kernforschung m.b.H., Karlsruhe



## Zusammenfassung

Bei zweidimensionalen SN-Rechnungen lassen sich Rechenzeitgewinne erzielen, wenn man die Anfangsflußschätzung mit Hilfe einer SN-Rechnung mit weniger Ortspunkten bereitstellt. Auch dieser "Vorlauf" läßt sich in der Rechenzeit verkürzen, wenn man einen weiteren SN-Lauf vorschaltet. Es entsteht in diesem Zusammenhang die Frage nach Strategien, die die Rechenzeit optimal gestalten. In dieser Arbeit werden insgesamt 120 verschiedene Transportprobleme, die mit Hilfe des zweidimensionalen SN-Programmes SNØW mit verschiedenen Strategien gerechnet wurden, ausgewertet. Gleichzeitig wird eine Theorie entwickelt, die die mittleren Rechenzeitgewinne approximativ wiedergibt. Anhand dieser Theorie wurden Strategien aufgefunden, die bis zu 60% Rechenzeitgewinne liefern.

## Abstract

The time of two dimensional SN-calculations can be shortened if the initial flux is prepared by a SN-calculation with less spatial points. Also this initial calculation can be shortened in time in the same way. In this connection there arises the question for a strategy, which influences the global calculation-time in an optimal way. In the following you find a statistical interpretation of 120 transport-problems, calculated by the two dimensional SN-code "SNØW" with different strategies. Simultaneously there is developed a theory, describing the calculation-time in dependence of the applied strategy. This theory made it possible to find strategies which save about 60% calculation-time in the mean.



E R R A T A

zum KFK 1361

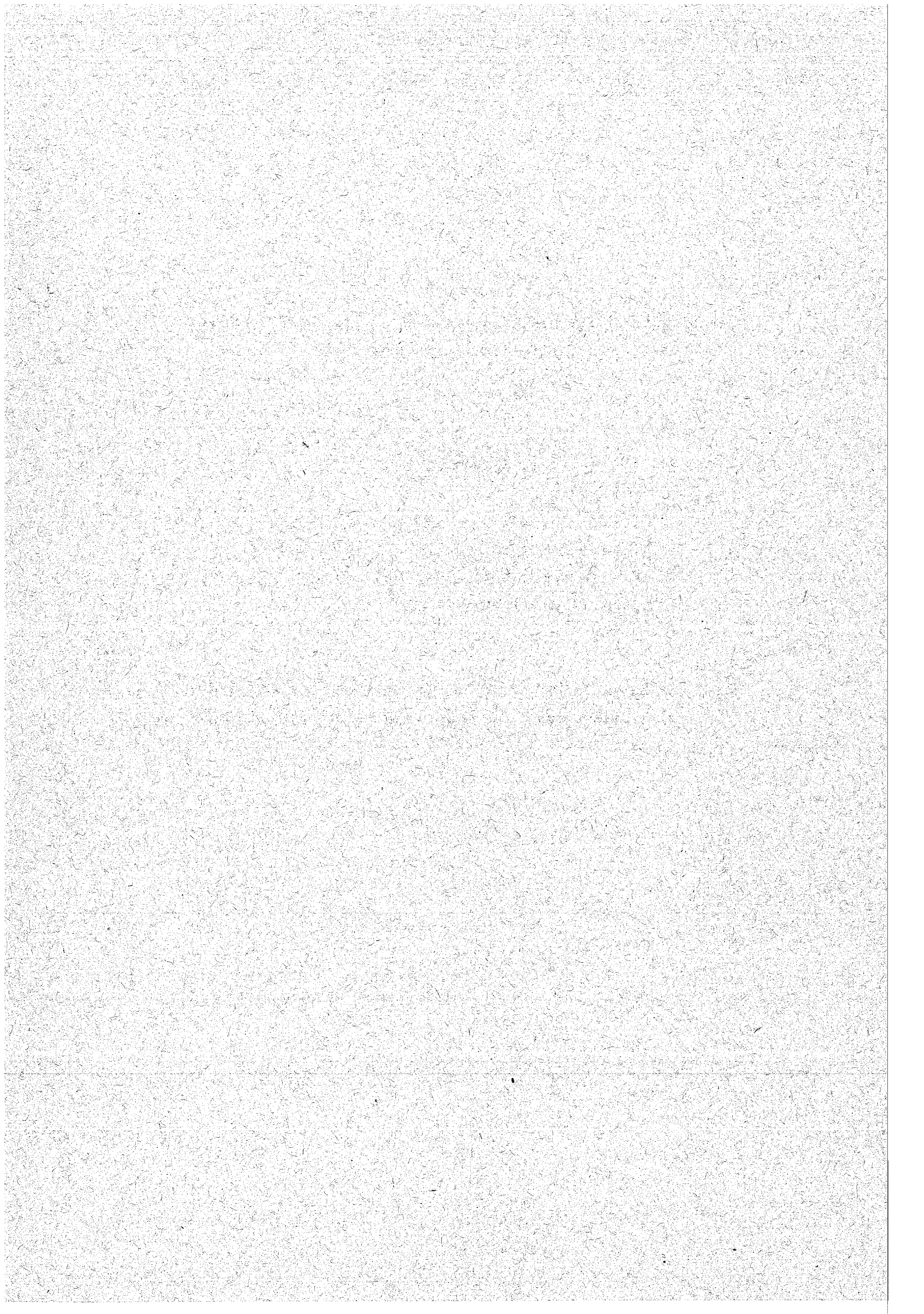
Strategien zur Beschleunigung zweidimensionaler SN-Verfahren  
mit Hilfe der Grobgittertechnik

von

W. Kinnebrock

Seite 2: In Formel (3) ist  $T\psi^{nn}$  durch  $T\psi^{n+1}$  zu ersetzen.  
In (4) und in der Formel der Zeile 17 ist  
 $\psi_j^{nn}$  durch  $\psi_j^n$  zu ersetzen.

Seite 12: In Zeile 13 ist der Satz "Im Gegensatz zur Strategie (6) ..." zu ersetzen durch "Im Gegensatz zur ersten Strategie ..."





## Einleitung

Mehrdimensionale SN-Programme benötigen neben großen Speicherkapazitäten auch hohe Rechenzeiten. Es ist daher unerlässlich, geeignete Beschleunigungsverfahren zu verwenden. Es gibt mehrere erfolgreiche Methoden, die Rechenzeit zu reduzieren, so z.B. die "Tschebyscheff-Beschleunigung" (5, 7) sowie das unter dem Namen "Rebalancing" bekannte Verfahren (3). Die Bereitstellung einer möglichst genauen Anfangsflußschätzung, etwa durch eine vorausgehende Difussionsrechnung, bringt ebenfalls große Rechenzeitgewinne. Schließlich ist es vorteilhaft, die ersten Iterationen einer SN-Rechnung wegen des in diesem Stadium des Programmablaufes noch recht ungenauen Flusses entweder mit einer groben Fehlerschätzung oder mit nur wenigen Gitterpunkten zu rechnen. Das letztere Verfahren wird z.B. in den Programmen DSN und TDC (2) angewendet, wo die erste äußere Iteration grundsätzlich als S2-Rechnung gerechnet wird. In dem eindimensionalen Programm DTFIV (3) werden die ersten inneren Iterationen nicht voll ausiteriert.

Eine erste Untersuchung über die Auswirkungen auf die Rechenzeit, wenn die Anfangsiterationen mit nur wenigen Gitterpunkten gerechnet werden, findet man bei Sullivan, Yu und Albers (10). Die Autoren führen die Anfangsiterationen mit einer niedrigen Anzahl von Winkeln durch. Der Rechenzeitgewinn beträgt bis zu 57%.

Das Ziel dieser Arbeit ist es, den Zeitgewinn experimentell und theoretisch zu ermitteln, wenn die ersten Iterationen mit wenigen Ortspunkten durchgeführt werden. Es wird versucht, optimale Strategien aufzufinden. Den experimentellen Ergebnissen liegen insgesamt 120 verschiedene SN-Rechnungen zugrunde, die mit Hilfe des zweidimensionalen Programmes SNØW (11) durchgeführt wurden. Es werden Strategien ermittelt, die durchschnittlich bis zu 60% Rechenzeitgewinn liefern.

---

Zum Druck eingereicht am 25. Januar 1971

1) Iterationszyklen und Rechenzeit bei SN-Verfahren

Die Lösung der stationären Boltzmann-Gleichung

$$(1) \quad \Omega \cdot \nabla \cdot \phi + \sigma_t \cdot \phi = S(\phi, K)$$

besitzt als Variablen den Winkel  $\Omega$ , den Raumvektor sowie die Energie  $E$ . Durch Hinzunehmen von Randbedingungen entsteht ein Eigenwertproblem mit dem Eigenwert  $K$ , welches durch geeignete Diskretisierung der unabhängigen Variablen in ein Matrix-Eigenwertproblem der Gestalt

$$(2) \quad T \psi = \frac{1}{K} F \psi + S \psi$$

übergeht. Im allgemeinen sind die Matrizen  $T, F, S$  so beschaffen, daß die Iteration

$$(3) \quad T \psi^{nn} = \frac{1}{K_n} F \psi^n + S \psi^{n+1}$$

mit  $K_n = (||F \psi^n||) / (||F \psi^0||)$  konvergiert (äußere Iteration). Die Matrix  $T$  ist eine Block-Diagonalmatrix mit den Blöcken  $T_{jj}$  und  $S$  ist (bei Fehlen von Aufwärtsstreuung) eine untere Block-Dreiecksmatrix mit den Blöcken  $S_{ij}$  ( $i > j$ ). Alle Matrizen  $T, S, F$  haben die gleiche Blockung.

Daher läßt sich (3) für  $r_n = \frac{1}{K_n} F \psi^n$  schreiben

$$T_{jj} \cdot \psi_j^{n+1} = S_{jj} \cdot \psi_j^{nn} + \sum_{i=1}^{j-1} S_{ji} \cdot \psi_i^{n+1} + r_n$$

$$\underline{j=1, 2, \dots, J; \quad \psi = (\psi_1, \psi_2, \dots, \psi_J)' \quad \underline{7}}$$

Die hier auftretenden Matrizen sind i.a. so komplex, daß eine direkte Berechnung von  $\psi_j^{n+1}$  nicht möglich ist. Das folgende - stets konvergierende - Iterationsschema mit dem Iterationsindex  $p$  erlaubt jedoch die Berechnung der Vektoren  $\psi_j^{n+1}$ :

$$(4) \quad T_{jj} (\psi_j^{n+1})^{p+1} = S_{jj} (\psi_j^{nn})^p + \sum_{i=1}^{j-1} S_{ji} \psi_i^{n+1} + r_n$$

$$(j=1, 2, \dots, J)$$

Für  $j=1$  fehlt der Summationsterm. Obige Iterationen heißen "innere Iterationen".

SN-Programme lösen das zu (1) konsistente Matrix-Eigenwertproblem (2) mit Hilfe von äußeren und inneren Iterationen. Wir werden im folgenden nur zweidimensionale Programme betrachten, welchen zwei Raum- und zwei Winkelvariablen sowie eine Energievariable zugrunde liegen. Programme dieser Art besitzen wegen der hohen Komponentenzahl des Vektors  $\psi$  große Rechenzeiten. So sind z.B. bei einer  $S_4$ -Rechnung mit 26 Energiegruppen und  $60 \times 60$  Ortsintervallen bei einer diskreten Winkelverteilung von 16 Richtungen auf der Halbkugel (vgl. [4, 7]) bei jeder inneren Iteration etwa  $1,5 \cdot 10^6$  verschiedene Flußwerte zu berechnen.

Es soll nun die Rechenzeit in Abhängigkeit von der Zahl der Ortspunkte dargestellt werden. Dazu machen wir die beiden folgenden - durch praktische Erfahrungen erhärteten - Annahmen.

- I) Die Gesamtanzahl  $I$  der inneren Iterationen bei einer SN-Rechnung ist wesentlich von der Anfangsschätzung  $\psi^0$  und der für die Iterationszyklen verwendeten Fehlergenauigkeit  $\epsilon$  und nur unwesentlich von den Schrittlängen des räumlichen Gitters abhängig.
- II) Die Rechenzeit ist direkt proportional der in dieser Rechnung ausgeführten Anzahl innerer Iterationen  $I$ , multipliziert mit der Dimension des Vektors  $\psi$ .

Es sei nun eine SN-Rechnung mit  $p \times q$  Ortspunkten und  $G$  Energiegruppen durchgeführt mit einer Rechenzeit  $T$ . Nach I) und II) ist

$$T = C \cdot I \cdot p \cdot q \quad (C > 0)$$

Eine SN-Rechnung mit der gleichen Gruppenzahl und Quadraturordnung, jedoch mit  $(\alpha \cdot p) \cdot (\alpha \cdot q)$  Ortspunkten ( $0 < \alpha < 1$ ) benötigen die Rechenzeit  $t(\alpha)$ . Dann gilt wegen I, II:

$$t(\alpha) = C \cdot I \cdot (\alpha \cdot p) \cdot (\alpha \cdot q)$$

d.h.

$$(5) \quad t(\alpha) = \alpha^2 \cdot T$$

## 2) SN-Rechnungen mit einem Vorlauf

Der Anfangsfluß  $\psi^0$  ist oft eine über alle Gitterpunkte konstante Funktion, günstigstenfalls eine die Randbedingungen berücksichtigende Parabel oder Sinus-Funktion.

Eine gute Anfangsschätzung erhält man, wenn man eine SN-Rechnung mit nur wenigen Ortspunkten vorschaltet (Vorlauf), etwa mit  $(\alpha \cdot p) \times (\alpha \cdot q)$  Punkten für  $0 < \alpha < 1$ , falls der eigentliche Lauf mit  $p \times q$  Punkten gerechnet werden soll. Wegen (5) ist die Rechenzeit eines Vorlaufes für kleine  $\alpha$  nur gering.

Es sei  $T$  die Zeit einer Rechnung mit  $p \times q$  Punkten ohne Vorlauf und  $t_2(\alpha)$  die Zeit einer Rechnung, deren Anfangsfluß  $\psi^0$  durch einen Vorlauf mit  $(\alpha \cdot p) \times (\alpha \cdot q)$  Punkten gewonnen wurde. Offensichtlich gilt

$$\begin{aligned} t_2(0) &= T \\ t_2(1) &= 0 \\ t_2'(\alpha) &< 0 \end{aligned}$$

Wir approximieren  $t_2(\alpha)$  durch

$$(6) \quad t_2(\alpha) = T \cdot (1-\alpha)^K$$

Die Zahl  $K$  ist abhängig von dem zugrunde liegenden Programm. Wir wählen  $K$  so, daß (6) optimale Werte für das zweidimensionale SN-Programm SNØW (vgl. [11\_7]) liefert. Es wurden insgesamt zehn verschiedene physikalische Probleme mit SNØW gerechnet, jeweils mit Vorläufen für mehrere  $\alpha$ . Die statistische Mittelung der Rechenzeiten  $t_2(\alpha)$  ergab für SNØW einen optimalen Wert von  $K=2$ .

Die gesamte Rechenzeit, die ein SN-Lauf mit einem Vorlauf von  $(\alpha \cdot p) \times (\alpha \cdot q)$  Punkten benötigt, ist nunmehr nach (5) und (6)

$$R(\alpha) = \alpha^2 \cdot T + T(1-\alpha)^2$$

Wir bezeichnen im folgenden als "relative Rechenzeit" den Wert  $\gamma(\alpha) = R(\alpha)/T$ . Es ist

$$(7) \quad \gamma(\alpha) = \alpha^2 + (1-\alpha)^2$$

Offensichtlich ist  $\gamma(\alpha)$  für  $\alpha=0,5$  minimal mit  $\gamma(0,5) = 0,5$ . Das bedeutet, daß bei einem Vorlauf mit  $\alpha=0,5$  ein Rechenzeitgewinn von 50% vorliegt.

Zur Nachprüfung dieses theoretisch ermittelten Wertes wurden mit Hilfe von SNØW zehn verschiedene Transportprobleme mit Vorläufen für verschiedene  $\alpha$  gerechnet. In Tabelle 1 sind die zehn gerechneten Anordnungen dargestellt.

Nr.	Bezeichnung	Literatur	Geometrie	SN-Ordnung	Punkte	Gruppen
1	SUAK U1B	[9]7	(x,y)	4	256	26
2	SNEAK 3A1	[6]7	(R,Z)	4	3696	2
3	TWØTRAN-SAMPLE PR.	[8]7	(x,y)	4	100	2
4	Krit.Anordnung (gemessen)		(R,Z)	4	506	26
5	wie 4, nur adjung.		(R,Z)	4	506	26
6	Unendl.Zylinder U235		(R,Ø)	4	80	2
7	DTF-SAMPLE-PR.	[3]7	(x,y)	4	324	3
8	Unendl.Zylinder		(R,Z)	4	150	8
9	wie 8		(R,Z)	8	150	8
10	wie 8, andere Abfragetechnik		(R,Z)	4	150	8

Tabelle 1

Für jede dieser Anordnungen wurden Rechnungen durchgeführt mit Vorläufen für  $\alpha = 0,4; 0,5; 0,6; 0,7; 0,8$ . Die relativen Rechenzeiten  $\gamma(\alpha)$  sind in Tabelle 2 gelistet (N = Nummer der Anordnung (vgl. Tab. 1)).

N \ $\alpha$	0,4	0,5	0,6	0,7	0,8
1	0,52	0,61	0,58	0,73	0,77
2	0,67	0,58	0,68	0,67	0,77
3	0,56	0,52	0,48	0,57	0,70
4	0,68	0,48	0,55	0,52	0,65
5	1,00	0,82	0,94	0,66	0,80
6	0,62	0,40	0,39	0,75	0,55
7	0,65	0,66	0,65	0,66	0,79
8	0,40	0,19	0,21	0,25	0,41
9	0,28	0,32	0,33	0,37	0,53
10	0,56	0,41	0,43	0,62	0,60

Tabelle 2

Man sieht, daß die Ergebnisse stark von den Anordnungen abhängen, jedoch gilt allgemein, daß  $\alpha = 0,5$  optimale Rechenzeiten liefert. In Tabelle 3 ist in Spalte 2 der arithmetische Mittelwert über alle Anordnungen für jedes  $\alpha$  eingetragen. In Spalte 3 finden sich die mit Hilfe von (7) errechneten relativen Rechenzeiten.

$\alpha$	gemittelte Zeiten	errechnete Zeiten
0,4	0,59	0,52
0,5	0,50	0,50
0,6	0,52	0,52
0,7	0,58	0,58
0,8	0,66	0,68

Tabelle 3

### 3) SN-Rechnungen mit mehreren Vorläufen

Es seien mehrere SN-Rechnungen mit den Punkten  $(\alpha_j \cdot p) \times (\alpha_j \cdot q)$  mit  $0 < \alpha_1 < \alpha_2 < \dots < \alpha_n < 1$  hintereinandergeschaltet. Der Ausgangsfluß einer Rechnung diene bei der folgenden (nach einer linearen Interpolation) als Anfangsfluß. Wenn  $t(\alpha_j, \alpha_{j+1})$  die Rechenzeit bedeutet, die der j-te Lauf benötigt, so gilt (vgl. Kap. 2)):

$$\begin{aligned}t(\alpha, \alpha) &= 0 \\t(0, \alpha) &= \alpha^2 \cdot T \\t(\alpha, 1) &= T \cdot (1-\alpha)^2\end{aligned}$$

Wir setzen daher

$$(8) \quad t(\alpha_j, \alpha_{j+1}) = T(\alpha_{j+1} - \alpha_j)^2$$

Die relative Rechenzeit bei einer Rechnung mit den Vorläufen mit  $0 < \alpha_1 < \alpha_2 < \dots < \alpha_n < 1$  gegenüber einer normalen Rechnung ist dann

$$(9) \quad \gamma(\alpha_1, \alpha_2, \dots, \alpha_n) = \left( \sum_j t(\alpha_j, \alpha_{j+1}) \right) / T = \sum_{j=0}^n (\alpha_{j+1} - \alpha_j)^2$$

für  $\alpha_0 = 0$  und  $\alpha_{n+1} = 1$ .

Speziell ergibt sich für

$$\alpha_j = \frac{j}{n+1} \quad (j=1, 2, \dots, n)$$

die relative Rechenzeit:

$$\gamma(\alpha_1, \dots, \alpha_n) = \frac{1}{n+1}$$

d.h. für wachsendes  $n$  wird die Rechenzeit beliebig klein. Letzteres steht im Widerspruch zur Erfahrung. Der Grund liegt in den beiden folgenden Effekten:

1. Die Voraussetzung  $t(\alpha, \alpha) = 0$  ist theoretisch. In der Praxis wird bei einem durch  $t(\alpha, \alpha)$  beschriebenen Lauf mindestens eine äußere Iteration durchgeführt mit allen zugehörigen peripheren Programmdurchläufen bis zu der Feststellung, daß Eingangs- und Ausgangsfluß identisch sind. Die hierdurch verursachte Rechenzeitverzögerung ist proportional  $\alpha^2$ .
2. Nichtberücksichtigt im obigen Modell sind die Interpolationszeiten, die für kleine  $n$  zwar vernachlässigbar, für große  $n$  jedoch ins Gewicht fallen. Auch diese sind proportional zu  $\alpha^2$ .

Es ist daher notwendig, die Formel (9) zu korrigieren, etwa durch Anhängen eines Terms

$$C \cdot \sum_{j=1}^{n-1} \alpha_j^2 \quad (C > 0)$$

Die in 1) beschriebene Rechenzeitverzögerung tritt in der Praxis dann auf, wenn in einem Zwischenlauf  $\alpha_j$  und  $\alpha_{j+1}$  sich nur wenig unterscheiden, so daß eine mit  $C \cdot \alpha^2$  korrigierte Formel dann gute Rechenzeiten liefert, wenn die Anzahl der Vorläufe groß ist.

Wir setzen also

$$(10) \quad \gamma(\alpha_1, \alpha_2, \dots, \alpha_n) = \sum_{j=0}^n (\alpha_{j+1} - \alpha_j) + C \cdot \sum_{j=0}^{n-1} \alpha_j^2$$

Diese Formel ist, wie erwähnt, um so genauer, je größer  $n$  ist. Die Konstante  $C$  ist programmabhängig. Für SNØW erhält man experimentell  $C = 0,25$ .

Die mit (10) für verschiedene Strategien errechneten Rechenzeiten sollen im folgenden mit experimentellen Ergebnissen verglichen werden. Die Anordnungen der Tabelle 1 wurden mit Hilfe verschiedener Strategien mit zwei, drei und vier Vorläufen gerechnet. In Tabelle 4 sind die den ausgewählten Strategien zugeordneten  $\alpha_v$  ( $v=1,2,\dots$ ) sowie die relativen Rechenzeiten der Anordnung 1 bis 10 gelistet. Wie man sieht, scheint die Vorlauf-Kombination mit

$$\alpha_1 = 0,13; \alpha_2 = 0,25; \alpha_3 = 0,50$$



Anzahl der Vorläufe	<u>Strategie</u>				<u>A n o r d n u n g</u>									
	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	1	2	3	4	5	6	7	8	9	10
2	0,33	0,66			0,62	0,48	0,50	0,43	0,57	0,59	0,58	0,30	0,29	0,48
2	0,25	0,50			0,57	0,41	0,44	0,48	0,49	0,32	0,60	0,27	0,25	0,46
2	0,50	0,75			0,64	0,55	0,44	0,50	0,75	0,38	0,70	0,23	0,34	0,42
3	0,25	0,50	0,75		0,62	0,48	0,50	0,45	0,56	0,57	0,65	0,34	0,28	0,47
3	0,13	0,25	0,50		0,60	0,40	-	0,48	0,48	0,37	0,61	0,31	0,28	0,50
4	0,20	0,40	0,60	0,80	0,69	0,48	0,49	0,48	0,61	0,77	0,70	0,26	0,29	0,56

Tabelle 4

die optimalen Rechenzeiten zu liefern. Arithmetische Mittelung über alle Anordnungen ergibt die in Tabelle 5 aufgezeichneten relativen Rechenzeiten. Zum Vergleich sind in der letzten Spalte die mit (10) errechneten Zeiten aufgetragen.

Anzahl der Vorläufe	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	gemittelte Zeiten	errechnete Zeiten
2	0,33	0,66			0,49	0,47
2	0,25	0,50			0,43	0,45
2	0,50	0,75			0,50	0,58
3	0,25	0,50	0,75		0,49	0,47
3	0,13	0,25	0,50		0,45	0,43
4	0,20	0,40	0,60	0,80	0,53	0,50

Tabelle 5

#### 4) Optimale Strategien

Wie bereits erwähnt, ist die Formel (10) für große  $n$  gut geeignet, da die ihr zugrunde liegenden Annahmen teilweise erst bei einer großen Anzahl von Vorläufen voll gerechtfertigt sind. Es ist daher möglich, mit Hilfe von (10) zu untersuchen, wie das Vorschalten einer großen Anzahl von Vorläufen vor den Hauptlauf sich in der Rechenzeit auswirkt.

Im folgenden werden zwei Strategien, die sich bei den Untersuchungen in Kap. 3) als günstig herausstellten, für  $n=2,3,4,\dots,10$  betrachtet. Die erste ist gegeben durch die  $\alpha$ -Werte:

$$\alpha_j = j/(n+1) \quad (j=1,2,\dots,n)$$

Für  $n=2,3$  erhält man eine relative Rechenzeit von 0,47 (vgl. Tabelle 5). Tabelle 6 zeigt die Rechenzeiten für alle  $n \leq 10$ :

n	errechnete Zeiten
2	0,47
3	0,47
4	0,50
5	0,55
6	0,61
7	0,67
8	0,74
9	0,81
10	0,89

Tabelle 6

Für  $n = 2,3$  erhält man die optimalen Rechenzeiten.

Die zweite zu betrachtende Strategie sei gegeben durch die  $\alpha$ -Werte

$$(11) \quad \alpha_j = 1/2^{n+1-j} \quad (j=1,2,\dots,n)$$

Tabelle 7 zeigt die relativen Zeiten für  $2 \leq n \leq 10$

n	errechnete Zeiten
2	0,4531
3	0,4258
4	0,4190
5	0,4050
6	0,4046
7	0,4045
8	0,4045
9	0,4045
10	0,4045

Tabelle 7

Diese Strategie ist offensichtlich für große  $n$  günstig. Im Gegensatz zur Strategie (6) konvergieren die Zeiten für wachsendes  $n$  gegen einen festen Wert. Für eine große Anzahl von Vorläufen gewinnt man etwa 60% an Rechenzeit.

Um zu prüfen, ob Strategien mit besserer Rechenzeitausnutzung existieren, bilden wir die ersten Ableitungen der Funktion  $\gamma(\alpha_1, \alpha_2, \dots, \alpha_n)$  in (10) und setzen

$$\frac{\partial \gamma}{\partial \alpha_1} = \frac{\partial \gamma}{\partial \alpha_2} = \dots = \frac{\partial \gamma}{\partial \alpha_n} = 0$$

Man erhält ein Gleichungssystem für  $\alpha_1, \alpha_2, \dots, \alpha_n$ . Die Lösung ergibt die bestmögliche Strategie. In Tabelle 8 sind die so bestimmten optimalen Strategien für  $n=2,3,4,5,6,7$  aufgezeichnet. Die letzte Spalte enthält die zugehörigen relativen Rechenzeiten:

n	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	errechnete Zeiten
2	0,25	0,55						0,4459
3	0,15	0,33	0,59					0,4105
4	0,09	0,16	0,33	0,59				0,4003
5	0,04	0,09	0,16	0,33	0,59			0,3967
6	0,02	0,05	0,10	0,20	0,36	0,60		0,3926
7	0,01	0,03	0,06	0,12	0,22	0,37	0,61	0,3913

Tabelle 8

Der Vergleich mit Tabelle 7 zeigt, daß die durch (11) definierte Strategie fast die best möglichen Zeiten erreicht.

Man kann also zusammenfassend feststellen, daß bei einem SN-Problem mit  $pxq$  verschiedenen Ortspunkten die Vorlaufkombination

$$\begin{aligned}
 & p/2 \times q/2 \\
 & p/4 \times q/4 \\
 & \dots\dots\dots \\
 & p/2^n \times q/2^n
 \end{aligned}$$

einen Rechenzeitgewinn von knapp 60% liefert. Die Zahl  $n$  ist dabei beschränkt durch  $p$ ,  $q$  sowie durch die Anzahl der Reaktorzonen, da im ersten Vorlauf in jeder Zone mindestens ein Intervall liegen sollte. Diese Strategie eignet sich insbesondere zum standardmäßigen Einbau in das Programm.

Literatur

- [1] B.G. Carlson  
Solution of the transport equation by SN-approximation  
LA-1891 (1955)
- [2] B.G. Carlson, C.E. Lee, W.J. Worlton  
The DSN and TDC neutron transport codes  
LA-2346 (1959)
- [3] K.D. Lathrop  
DTF IV, a Fortran IV program for solving the multigroup  
transport equation with anisotropic scattering  
LA-3373 (1965)
- [4] K.D. Lathrop, B.G. Carlson  
Discrete ordinates angular quadrature of the neutron  
transport equation  
LA-3186 (1965)
- [5] R.S. Varga  
Matrix iterative analysis
- [6] L. Barleon et al.  
Physics investigations of a 6701 steam cooled fast reactor  
system in SNEAK, assembly 3A-1  
KFK 627 (1967)
- [7] K.D. Lathrop  
User's guide for the TWØTRAN (x,y) program  
LA-4058 (1968)
- [8] K.D. Lathrop  
TWØTRAN - a FØRTRAN program for two dimensional transport  
G.A. 8747 (1968)
- [9] F. Mitzel, K.E. Schroeter  
Gepulste Experimente an der Schnellen Unterkritischen Anlage  
(SUAK) und Vergleich der Ergebnisse mit Experimenten  
Nukleonik 12, 110 - 117 (1969)
- [10] R.E. Sullivan, D. Yu, L.K. Albero  
Reduction of machine time in discrete ordinates calculations  
Proc. of conf. on the effective uses of comp. in the nucl.indust.  
Knoxville, Tennessee (1969)
- [11] C. Günther, W. Kinnebrock  
SNØW - Ein zweidimensionales SN-Programm zur Lösung der Neutronen-  
Transportgleichung in Platten- und Zylindergeometrie  
KFK, erscheint demnächst