

**MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK**  
**GARCHING BEI MÜNCHEN**

Benchmark-Versuche mit der CRAY-1

Friedrich Hertweck  
Wolfgang Schneider  
Ulrich Schwenn

IPP R/31  
1/170  
6/184

Mai 1979

*Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem  
Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die  
Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.*

## Benchmark-Versuche mit der CRAY-1

Friedrich Hertweck  
Wolfgang Schneider  
Ulrich Schwenn

|  |    |
|--|----|
| 1. Einleitung                                      | 1  |
| 1.1 Die CRAY-1                                     | 1  |
| 1.2 Computational Physics im IPP                   | 1  |
| 2. Die Benchmarks                                  | 3  |
| 2.1 Erfahrungen mit dem CRAY-System im RHEL        | 3  |
| 2.2 Tests mit einfachen Programmkernen             | 4  |
| 2.2.1 Einfache Schleifen                           | 4  |
| 2.2.2 Komplexere Schleifen                         | 5  |
| 2.3 Die "Number-Cruncher" des IPP                  | 6  |
| 2.3.1 Programm BRM                                 | 6  |
| 2.3.2 Programm UHS                                 | 7  |
| 2.3.3 Programm ROC                                 | 8  |
| 2.3.4 Programm HEW                                 | 9  |
| 2.3.5 Programm MCK                                 | 10 |
| 2.4 Optimierungsversuche                           | 10 |
| 2.4.1 Optimierung des Programms BRM                | 11 |
| 2.4.2 Optimierung des Programms UHS                | 11 |
| 3. Folgerungen                                     | 13 |
| Appendix 1: Leistungsvergleich: Compiler           | 14 |
| Appendix 2: Leistungsvergleich: MHD-Code Schleifen | 15 |
| Appendix 3: Leistungsvergleich: MHD-Code Programme | 16 |

## 1. Einleitung

Im Rahmen der Ueberlegungen, durch welchen Rechner die IBM 360/91 des IPP ersetzt werden soll, wurden auch Benchmark-Versuche mit der CRAY-1 durchgefuehrt. Die Ergebnisse werden im folgenden beschrieben. Die Vergleiche beziehen sich auf beide Rechner des IPP, die IBM 360/91 und die AMDAHL 470V/6.

### 1.1 Die CRAY-1

Die CRAY-1 ist ein leistungsfaehiger Rechner fuer wissenschaftliche Anwendungen. Sie hat eine CPU-Zykluszeit von 12.5 nsec und eine Speicherzykluszeit von 50 nsec. Sie ist sowohl fuer (auch auf anderen Rechnern uebliche) Skalar- als auch fuer Vektor-Operationen konstruiert worden. Der Speicher kann maximal 1 Million Worte (@ 64 bits) haben. Er besteht aus 16 Baenken und ist in ECL-Technologie mit SEC/DED (Single Error Correction / Double Error Detection) ausgefuehrt. Der Speicher kann 80 Mio Worte/sec mit der CPU austauschen.

Die CPU hat 12 Funktionseinheiten, die unabhaengig voneinander arbeiten koennen (address, add/multiply, scalar add/shift/logical/bit count, vector add/shift/logical, floating point add/multiply/divide), ein 7-bit Register, 72 24-bit Register, 73 64-bit Register, 8 vector Register (64 Worte @ 64 bits). Die Vector-Funktionseinheiten fuer Addition und Multiplikation koennen jede 12.5 nsec ein Ergebnis liefern, die Divisionseinheit jede 37.5 nsec ein Ergebnis. Man erreicht damit theoretisch maximal 185 Mio Operationen/sec.

Die CRAY-1 ist physisch sehr kompakt (Rechner + Speicher: 144 cm Durchmesser, Bodenflaeche einschl. Netzgeraete: 263 cm Durchmesser) und ist Freon-gekuehlt (d.h. sie tauscht keine Waerme mit der umgebenden Luft aus). Sie verbraucht bei maximalem Speicherausbau 115 KW.

### 1.2 Computational Physics im IPP

Ein wesentlicher Beitrag zur theoretischen Beschreibung des Plasmas wird von MHD-Modellen (Einwirkung von elektromagnetischen Feldern auf das Plasma als Fluessigkeit) geleistet.

Alle diese Modelle, die nicht nur eine umfassende Behandlung der verschiedenen Plasmaeffekte, sondern auch realistische Geometrien und Randbedingungen behandeln, koennen nur numerisch geloest werden. Die Komplexitaet der Problemstellung hat nicht nur zu grossen Codes (300-500 KBytes Code und derzeit 2000-3000 KBytes Datenfelder) sondern insbesondere auch zu grossen Rechenzeiten (20-50 Std. fuer einen Parameterfall) gefuehrt.

Eine etwas willkuerliche Einteilung der Problemkreise ist die folgende:

- (1) Das fundamentalste Problem ist die Berechnung von MHD-Gleichgewichten. Fuer Konfigurationen mit einer Symmetrie (Axialsymmetrisch: z.B. Tokamak; helikalsymmetrisch: z.B. gerader Stellarator) kann das Problem als numerische Randwertaufgabe geloest werden. Ein solches 2D-Gleichgewichtsprogramm (BRM) wurde auch fuer die Benchmarks herangezogen.

Fuer Konfigurationen ohne Symmetrieebene (Stellarator) muessen fuer das Auffinden von Gleichgewichten andere Wege beschritten werden: entweder durch die Loesung der MHF-Gleichungen (Magneto-Hydro-Friction) (Programm ROC) oder durch die Anwendung eines Energieprinzips fuer die idealen MHD-Gleichungen (Programm UHS). Diese 3D Programme sind im IPP soweit entwickelt worden, dass zur Zeit fuer das weitere Vorgehen die Rechenkapazitaet eine stark einschraenkende Bedingung darstellt.

Diese Modelle liefern ueber die Gleichgewichtsloesung hinaus auch Aussagen ueber die Stabilitaet der Konfiguration.

- (2) Fuer die 2D Konfigurationen (z.B. Tokamak) sind in den letzten Jahren immer genauere Methoden fuer die numerische Modenanalyse entwickelt worden. Die zunaechst dafuer entwickelten Anfangswertproblem-Codes werden zur Zeit wenig genutzt, da neuere Methoden (finite elements) und Programme (ERATO) die genaueren Resultate liefern. Auch diese 2D Stabilitaetsrechnungen sind sehr rechenintensiv. Aus technischen Gruenden (siehe 2.3) wurden diese Programme jedoch nicht fuer die Benchmarks verwendet.

Die eben beschriebene Methode gibt Resultate nur fuer 2D Konfigurationen und ideale MHD.

Fuer die 3D-Konfigurationen erhaelt man aus den 3D-Modellen (vgl. 1) Aussagen ueber die Stabilitaet in idealer MHD.

Fuer 2D Konfigurationen und nicht ideale MHD (z.B. endlicher elektrischer Widerstand) fuehrt das Stabilitaetsproblem auf ein 3D-Problem. Als Beispiel dieser Arbeiten wurde das Programm HEW fuer die Benchmarks herangezogen.

- (3) Ein weiterer wichtiger Problemkreis ist die Beschreibung des Transportverhaltens des Plasmas (also nichtideale MHD). Diese Rechnungen behandeln sehr detaillierte Plasmamodelle (z.B. Einschuss von neutralen Teilchen, Interkorrelation von Verunreinigungen etc.). Ein Teil dieser Untersuchungen ist enthalten im Programm MCK, das als fuenftes Beispiel der "Grossen" fuer die Benchmarks verwendet wurde.

## 2. Die Benchmarks

In diesem Abschnitt werden einige Benchmark-Versuche beschrieben. Zum Teil wurden sie mit synthetischen Programmen ausgeführt, zum Teil mit Produktionsprogrammen des IPP. Die Tests wurden auf der CRAY-1 ausgeführt, die (im Besitz der Firma Cray Research) im Rutherford High Energy Laboratory (RHEL) vorübergehend aufgestellt ist. Die Anlage hat 512K Worte Arbeitsspeicher, einige Plattenspeicher und als Interface zur Aussenwelt einen Minirechner Eclipse 200; dieser dient als Wartungskonsole und ist mit einem kleinen Plattenspeicher, einem Bandgeraet, einem Kartenleser und einem Zeilendrucker ausgeruestet.

### 2.1 Erfahrungen mit dem CRAY-System im RHEL

Alle ausgewaehlten Programme aus dem IPP wurden auf Magnetband kopiert. Nach anfaenglichen Schwierigkeiten (die IPP-Bandeinheit war nicht richtig justiert) konnte das von den Autoren mitgebrachte Band ohne Probleme gelesen werden - die Programme standen zur Verfuegung.

Moeglichkeiten fuer Editing und File-handling waren durch zwei Terminals gegeben. Der Eclipse-Minirechner mit Plattenspeicher uebernimmt das Editing und das Erstellen von CRAY-Files (diese muessen auf spezielle Weise geblockt werden). Begonnen wurde mit dem Programm BRM, das innerhalb von 45 min in vollem Umfang lief. Bei den laengeren Codes zeigten sich bald die Grenzen des Minirechners (nicht der CRAY-1!). Bei rund 7000 Karten des ROC-Programms dauerte eine Aenderung des Gesamt-Files ca. 8 min. Nach insgesamt ca. 20 Mannstunden liefen alle Jobs, das Optimieren konnte beginnen.

Waehrend der gesamten Testzeit ereignete sich ein Zusammenbruch des Eclipse-Rechners und ein Hardwarecheck der CRAY-1. In beiden Faellen war das System nach weniger als einer Minute wieder betriebsbereit.

Es wurden keine Tests fuer I/O-Vorgaenge gemacht, die Aussagen ueber den Datenfluss zwischen CRAY-1 und externen Speichermedien zulassen.

Die Beschraenkungen beim Editing fallen natuerlich fort, wenn die CRAY-1 an einen Vorrechner (z.B. die Amdahl) gekoppelt wird. Die Firma Cray Research liefert dazu die Hardware und die Software. Dann koennen alle Programme auf dem Vorrechner editiert werden.

## 2.2 Tests mit einfachen Programmkernen

In diesem Abschnitt werden Tests mit einfachen Programm-Kernen beschrieben.

### 2.2.1 Einfache Schleifen

Um die Leistung des Rechners fuer typische Vektorschleifen zu testen, wurden einfache Schleifen, bestehend aus einem FORTRAN-Statement, fuer Vektoren der Laenge 100 wiederholt (meist 1000000fach) ausgefuehrt. Es handelte sich um folgende Schleifen:

```
DO 1 N=1,100
1   Z(N)=X(N)+Y(N)

DO 2 N=1,100
2   Z(N)=X(N)*Y(N)

DO 2 N=1,100
3   Z(N)=X(N)/Y(N)

DO 4 N=1,100
4   Z(N)=(X(N)-Y(N))*(X(N)+Y(N))

S=0.0
DO 5 N=1,100
5   S=S+X(N)*Y(N)

DO 6 N=1,100
6   Z(N)=0.5*(X(N)/Z(N) + Y(N)*Z(N))

DO 7 K=1,10
DO 7 N=1,100
7   Z(N)=Z(N)*X(N) + Y(K)

DO 8 N=2,100
8   Z(N)=X(N)-X(N-1)

DO 9 N=2,99
9   Z(N)=X(N+1) - 2.0*X(N) + X(N-1)
```

Die Ergebnisse sind in der folgenden Tabelle zusammengefasst.

| Programm-Kern        | Exec-Zeiten [sec] |        |        | Leistung |          |
|----------------------|-------------------|--------|--------|----------|----------|
|                      | CRAY-1            | 360/91 | 470V/6 | CRAY:91  | CRAY:V/6 |
| X+Y                  | 5.36              | 44     | 96     | 8.2      | 17.9     |
| X*Y                  | 5.39              | 43     | 133    | 8.0      | 24.6     |
| X/Y                  | 10.26             | 67     | 307    | 6.5      | 29.9     |
| (X-Y)*(X+Y)          | 6.91              | 67     | 202    | 9.7      | 29.3     |
| X.Y (Skalarprodukt)  | 6.10              | 39     | 124    | 6.4      | 20.3     |
| Z:=0.5*(X/Z+Y*Z)     | 13.74             | 132    | 488    | 9.6      | 35.5     |
| P(X) (Polynom)       | 59.81             | 475    | 1686   | 7.9      | 28.2     |
| X[i]-X[i-1]          | 5.36              | 68     | 123    | 12.7     | 22.9     |
| X[i+1]-2*X[i]+X[i-1] | 8.39              | 87     | 220    | 10.4     | 26.2     |
| Mittelwerte          |                   |        |        | 9.0      | 26.1     |

### 2.2.2 Komplexere Schleifen

Als naechster Schritt wurden drei typische, jeweils dreifach genestete Schleifen untersucht. Die erste definiert 3D Arrays; die zweite rechnet mit Hilfe finiter Differenzen eine neue Groesse aus; in der letzten Schleife wird eine Summation ueber das ganze Gitter durchgefuehrt.

Die Zeitmessungen fuer dieses kleine Testprogramm mit einfacher und doppelter Genauigkeit und auch mit abgeschalteter Vektorisierung ergaben folgende Zeiten (jeweils gemessen fuer einen verschieden grossen Bereich des 1. Index, d.h. der inneren Schleife):

|                 | Gitter   | IBM360/91 | 470V/6    | CRAY-1   |
|-----------------|----------|-----------|-----------|----------|
| REAL*4          | 12x12x12 | 2.95 sec  | 3.26 sec  | -        |
|                 | 72x12x12 | 20.50 sec | 20.79 sec | -        |
| REAL*8          | 12x12x12 | 2.85 sec  | 5.88 sec  | .48 sec  |
|                 | 72x12x12 | 21.50 sec | 34.69 sec | 1.94 sec |
| REAL*8 (scalar) | -        | -         | -         | 4.06 sec |

Dabei konnten die beiden ersten Schleifen durch den CRAY-Compiler "vektoriert" uebersetzt werden, waehrend die dritte Schleife nur in "skalaren" Code umgesetzt werden konnte.

Um den Einfluss der "Vektorisierung" auf die Rechenzeit zu erfassen, wurde das Programm zusaetzlich nur in skalarer Form ausgefuehrt, die Rechenzeit in dieser Form war dann wie in der letzten Zeile der obigen Tabelle angegeben.

Das Vektorisieren von 2 Schleifen erbrachte also in diesem Fall eine Halbierung der Rechenzeit.

Die Leistungsfaeohigkeit der CRAY (teilweise vektorisiertes Programm) im Vergleich mit den IPP-Rechnern ergibt damit folgendes Bild:

| REAL*8   | L[CRAY]<br>-----<br>L[91] | L[CRAY]<br>-----<br>L[470] |
|----------|---------------------------|----------------------------|
| 12x12x12 | 6.0                       | 12.3                       |
| 72x12x12 | 11.1                      | 17.9                       |

### 2.3 Die "Number-Cruncher" des IPP

Die Auswahl aus dem "Angebot" des IPP musste zwei Randbedingungen beruecksichtigen. Einmal zwang die an der CRAY-1 zur Verfuegung stehende Arbeitszeit zur Beschraenkung auf wenige Programme, die moeglichst ohne periphere Speicher auskommen. Dies trifft mit Ausnahme der Stabilitaetsrechnungen ERATO auf die Situation im IPP zu. CPU-intensive Jobs nutzen externe Speicher fast ausschliesslich fuer "Restart" und "Backup" Funktionen. Zum zweiten sollten Programme getestet werden, die fuer die Arbeiten des IPP typisch sind und einen erheblichen Anteil am Gesamtbedarf an Rechenzeit der IPP-Benutzer haben.

#### 2.3.1 Programm BRM

Das Programm berechnet 2D-MHD-Gleichgewichte (fuer helikale Symmetrie) in einem kartesischen Koordinatensystem. Als numerisches Verfahren wird ein "alternating direction implicit" Schema iterativ verwendet. Das Verfahren benoetigt doppelte Genauigkeit.

Die Zeitmessungen ergaben fuer ein 64x64 Gitter pro Iterationsschritt:

| REAL*8      | IBM 360/91 | AMDAHL 470/V6 | CRAY-1  |
|-------------|------------|---------------|---------|
| Compilation | 9.62 sec   | 5.22 sec      | .53 sec |
| Execution   | .62 sec    | .92 sec       | .11 sec |



Die Leistung der CRAY im Vergleich mit den IPP-Rechnern ist somit:

| REAL*8    | L[CRAY]<br>-----<br>L[91] | L[CRAY]<br>-----<br>L[470] |
|-----------|---------------------------|----------------------------|
| Compiler  | 18.2                      | 9.9                        |
| Execution | 5.7                       | 8.4                        |

Eine zusaetzliche Zeitmessung der nichtvektorierten Version ergab im Vergleich zur Standardversion:

$$\frac{L \text{ [CRAY: vektorisiert]}}{L \text{ [CRAY: scalar]}} = 1.06$$

d.h. ein grosser Teil der ausgefuehrten Instruktionen ist nicht vektorisiert. Aus diesem Grund wurde das Programm hinsichtlich seiner Vektorisierbarkeit spaeter optimiert (siehe 2.4).

### 2.3.2 Programm UHS

Hier handelt es sich um ein dreidimensionales Programm zur numerischen Berechnung MHD-stabiler Konfigurationen. Ausgehend vom Variationsprinzip fuer die idealen MHD-Gleichungen werden durch ein geeignetes Iterationsverfahren Zustaeude minimaler Energie bestimmt, die dann auf Stabilitaet untersucht werden. Das dreidimensionale Gitter umfasst derzeit typisch 8...16 Punkte in radialer-, 24...64 Punkte in poloidaler- und 16...64 Punkte in toroidaler Richtung. Das Maximum, bedingt durch den verfuegbaren Hauptspeicher auf der 470V/6 liegt bei 12x48x48 Punkten, auf der 360/91 entsprechend niedriger.

Zum Test auf der CRAY-1 wurde eine Version verwendet, die nur aus den CPU-intensiven Routinen bestand. Der Overhead durch Kontrollrechnungen, I/O-Operationen etc. liegt bei ca. 30 min Gesamtrechnzeit unter 5%, ermittelt auf der 470V/6 und der 91, so dass diese Massnahme zulaessig ist. Zur Vorbereitung wurde das Programm in Garching von doppelter auf einfache Genauigkeit umgestellt.

Die nicht Standard-FORTRAN entsprechende Behandlung variabler Dimensionen im Original-Code erforderte ca. 20 min. Editing und File-Handling auf der CRAY-1. Dabei mussten in mehreren Routinen je zwei COMMON und DIMENSION Statements vertauscht werden. Der erste Lauf mit einem Gitter von 12x32x16 Punkten lieferte folgende Ergebnisse, bezogen auf doppelte Genauigkeit auf 470V/6 und 91:

|                         | 360/91 | 470V/6 | CRAY-1 |
|-------------------------|--------|--------|--------|
| Compile-Zeit[sec]       | 19.1   | 10.5   | 0.638  |
| Zeit pro Iteration[sec] | 1.5    | 2.2    | 0.22   |

Daraus ergeben sich die Verhaeltnisse:

|           | L[CRAY]<br>-----<br>L[91] | L[CRAY]<br>-----<br>L[470] |
|-----------|---------------------------|----------------------------|
| Compiler  | 29.9                      | 16.5                       |
| Execution | 6.8                       | 10.0                       |

Diese Faktoren fuer die Rechengeschwindigkeit blieben auch bei Erhoehung der Punktzahl auf 12x32x32 und 12x32x64 erhalten. Zu bemerken ist noch, dass der optimale Wert fuer die 360/91 nur erreicht wurde, wenn umfangreichere Schleifen in eine Vielzahl kleinerer aufgespalten wurden, analog zu den in 2.2 untersuchten. Durch Abschalten des Vectormodes in der CRAY-1 erhoelte sich die Rechenzeit um den Faktor 1.48. Die Analyse des Flow-trace ergab, dass 55% der Rechenzeit in einem Unterprogramm verbraucht wurden. Dieses besteht aus einer Schleife mit einem sehr umfangreichen Kern, den der CRAY-Compiler zunaechst nicht vektorisieren konnte. Die Optimierung dieses Programms wird in 2.4 beschrieben.

### 2.3.3 Programm ROC

Dieses Programm berechnet mit Hilfe von finiten Differenzenmethoden 3D-Gleichgewichte in kartesischen Koordinaten.

Die Zeitmessungen fuer verschiedene Gitter ergeben (jeweils fuer einen Iterationsschritt):

| REAL*8               | IBM 360/91 | AMDAHL 470V/6 | CRAY-1    |
|----------------------|------------|---------------|-----------|
| Compilation          | 83.0 sec   | 40.0 sec      | 2.45 sec  |
| Execution            |            |               |           |
| 5x 5x 5              | .0235 sec  | .0480 sec     | .0059 sec |
| 10x10x10             | .1815 sec  | .3335 sec     | .0353 sec |
| 20x20x10             | .7077 sec  | 1.2954 sec    | .1254 sec |
| 20x20x10<br>(scalar) | -          | -             | .1558 sec |

Der Vergleich der vektorisierten Iteration mit der skalaren ergab

$$\frac{L \text{ [CRAY: vector]}}{L \text{ [CRAY: scalar]}} = 1.24$$

Vergleicht man die Leistungsfähigkeit der CRAY mit unseren Maschinen, so ergibt sich

| REAL*8    | L[CRAY]<br>-----<br>L[91] | L[CRAY]<br>-----<br>L[470] |
|-----------|---------------------------|----------------------------|
| Compiler  | 33.9                      | 16.3                       |
| Execution |                           |                            |
| 5x 5x 5   | 4.0                       | 8.1                        |
| 10x10x10  | 5.1                       | 9.5                        |
| 20x20x10  | 5.6                       | 10.3                       |

D.h. auch bei dem relativ geringen Ausmass an Vektorisierung ergeben laengere Vektoren einen groesseren Leistungsfaktor.

#### 2.3.4 Programm HEW

Dieses Programm wurde in den vergangenen Jahren intensiv zur Untersuchung der sog. "Tearing-Instabilitaeten" verwendet, einer wesentlichen Ursache der "disruptions" in Tokamak-Entladungen. Durch Differenzenverfahren werden die zeitabhaengigen resistiven MHD-Gleichungen in einem dreidimensionalen Gitter geloest. Die feine Struktur der physikalischen Vorgaenge erfordert entsprechende Gitter; typisch sind 96x32x16 Punkte. Dafuer werden 3 MBytes Hauptspeicher benoetigt, eine weitere Verfeinerung ist derzeit auch auf der Amdahl nicht moeglich. Dazu kommen durchschnittliche Rechenzeiten zwischen 10 und 20 Stunden auf der 470V/6.

Auch dieses Programm wurde in Garching vorbereitend auf einfache Genauigkeit umgestellt. Zusaetzlich mussten auf der CRAY-1 ein kombiniertes Integer-Data-Statement und einige doppeltgenaue Konstanten geaendert werden. Die Ergebnisse sind wie folgt:

|                         | 470V/6 | CRAY-1 |
|-------------------------|--------|--------|
| Compiler-Zeit[sec]      | 16.5   | 0.92   |
| Zeit pro Iteration[sec] | 30.0   | 7.65   |

Daraus ergeben sich Leistungsfaktoren von 17.9 fuer die Compilation und 3.92 fuer die Ausfuehrung.

Eine Analyse des Flow-Trace zeigte, dass 40% der Rechenzeit auf der CRAY-1 fuer Fourier-Transformationen verbraucht wurden, die der CRAY-Compiler nicht vektorisieren konnte. Die spezielle Programmierung der Randbedingungen durch eine Vielzahl von "IF"-Statements in den innersten Schleifen unterband auch im uebrigen Programm weitgehend die Vektorisierung der rechenintensiven Kerne. Eine Optimierung wurde nicht versucht, da zwar in der Subroutinen-Bibliothek der CRAY-1 eine Fourier-Routine vorhanden ist, aber der Umfang des Programms Eingriffe durch Aussenstehende in der

verfuegbaren Zeit nicht zuliess.

### 2.3.5 Programm MCK

Dieses in Princeton entwickelte Programm simuliert mit einem Monte-Carlo Verfahren die Absorption hochenergetischer Neutralteilchenstrahlen in Plasmen. Als Teil eines umfangreichen Transportcodes erlaubt es Vorhersagen und Vergleiche fuer gegenwaertige und zukuenftige Experimente. Diese Heizmethode fuer Fusionsplasmen wird von grosser Bedeutung in der Zukunft sein.

Die derzeitigen Beschraenkungen beruhen ueberwiegend auf dem extensiven Rechenzeitbedarf fuer die Teilchenbewegung, ueber 90% werden im Rahmen des Gesamttransportcodes dafuer verwendet.

Da das Programm mit dem OLYMPUS-System entwickelt und von CDC- auf IBM-Anlagen umgestellt worden war, ergaben sich einige notwendige Aenderungen, vor allem bei Bibliotheks-routinen; diese konnten jedoch problemlos durchgefuehrt werden. Folgende Werte wurden erzielt:

|                  | 470V/6<br>REAL*4 | 470V/6<br>REAL*8 | 360/91<br>REAL*4 | 360/91<br>REAL*8 | CRAY-1 |
|------------------|------------------|------------------|------------------|------------------|--------|
| Compilation[sec] | 17.0             | 17.0             | 27.0             | 27.0             | 0.76   |
| Execution[sec]   | 34.2             | 45.0             | 61.3             | 61.7             | 6.4    |

Die Leistungsfaktoren sind:

|                  | L[CRAY]<br>-----<br>L[91] | L[CRAY]<br>-----<br>L[470] |
|------------------|---------------------------|----------------------------|
| Compilation      | 35.6                      | 22.4                       |
| Execution REAL*4 | 9.6                       | 5.3                        |
| REAL*8           | 9.6                       | 7.0                        |

Zu beachten sind die Unterschiede zu den uebrigen Programmen. Der Aufbau eines solchen Monte-Carlo Programms erfordert wesentlich mehr logische Operationen als die Differenzenverfahren. (Die 470V/6 ist bei logischen Operationen leistungsfaehtiger als die 360/91; bei Floating Point Operationen ist es umgekehrt). Es zeigt sich, dass die CRAY-1 fuer beide Aufgaben eine sehr leistungsfaehtige Maschine ist, was vor allem auf die kurzen Zykluszeiten der CPU und auch des Speichers zurueckzufuehren ist.

### 2.4 Optimierungsversuche

Anhand von Messungen des Zeitverbrauchs in einzelnen Subroutinen ("Flow-Trace") konnten Schluesse gezogen werden, welche Schleifen durch Vektorisierung eine deutliche Performance-Verbesserung ergeben wuerden.

## 2.4.1 Optimierung des Programms BRM

Ein Hindernis fuer Vektorisierung zweier Schleifen im BRM-Programm war die Verwendung von skalaren Zwischen-groessen. Diese Groessen wurden als Vektoren geschrieben. Damit ergaben sich im Vergleich zur alten Version (2.3.1) und nicht vektorisiertem Code folgende Zeiten:

Programm BRM

| REAL*8 (64x64)     | CRAY-1              |
|--------------------|---------------------|
| Original-Version   | .1089 sec/iteration |
| optimiert (scalar) | .1157 sec/iteration |
| optimiert (vector) | .0634 sec/iteration |

D.h. die Optimierung ergab eine Leistungssteigerung von:

$$\frac{L \text{ [CRAY: vector]}}{L \text{ [CRAY: scalar]}} = 1.82$$

Vergleicht man diese optimierte Version mit den Zeitmessungen auf den IPP-Rechnern, so ergibt sich eine Steigerung der Leistungsfahigkeit der CRAY wie folgt:

| BRM                | L[CRAY]<br>L[91] | L[CRAY]<br>L[470] |
|--------------------|------------------|-------------------|
| nicht-optimiert    | 5.72             | 8.43              |
| optimiert (scalar) | 5.38             | 7.93              |
| optimiert (vector) | 9.82             | 14.48             |

Durch einen relativ einfachen Eingriff in das FORTRAN-Programm konnte die Ausfuehrungsgeschwindigkeit auf der CRAY-1 um fast einen Faktor 2 gesteigert werden.

## 2.4.2 Optimierung des Programms UHS

Die Ergebnisse mit dem 3-D Code UHS und die Analyse des Flow-Trace legten einen Optimierungsversuch nahe. Das Unterprogramm NEWWPP gehoert zum Kernteil des Codes und dient zur Berechnung der Energie zwischen zwei magnetischen Flaechen. Es besteht aus einer Schleife ueber 2 der 3 Ortsdimensionen. Der Kern enthaelt 8 Vektoroperationen und eine skalare Summe. Dazu kommen zwei "IF" Statements. Durch Aenderung der Anfangsindizes, d.h. getrennte Berechnung der Randbedingungen, entfielen letztere, durch eine zusaetzliche Schleife auch die Summation.

Dadurch ergab sich eine Verbesserung um den Faktor:

$$\frac{L \text{ [CRAY: vector]}}{L \text{ [CRAY: scalar]}} = 1.87$$

fuer das gesamte Programm (Gitter: 12x32x32).

Der Optimierungseffekt wird noch deutlicher, wenn man die durchschnittliche Rechenzeit fuer dieses Unterprogramm allein betrachtet. Um die Wirkung der Vektorisierung zu beurteilen, wurde auch bei der optimierten Version zusaetzlich der Vektor-Mode abgeschaltet:

CPU-Zeit je Aufruf von NEWWPP in msec:

| 470V/6<br>REAL*4 | 470V/6<br>REAL*8 | 360-91<br>REAL*4 | 360-91<br>REAL*8 | CRAY-1<br>(vector) | CRAY-1<br>(scalar) |
|------------------|------------------|------------------|------------------|--------------------|--------------------|
| 44.0             | 61.3             | 37.1             | 37.7             | 1.47               | 9.0                |

Damit ergeben sich folgende Leistungsfaktoren:

|             | L[CRAY]<br>-----<br>L[91] | L[CRAY]<br>-----<br>L[470] |
|-------------|---------------------------|----------------------------|
| Scalar mode | 4.2                       | 6.8                        |
| Vector mode | 25.6                      | 41.7                       |

Die Vektorisierung auf der CRAY-1 brachte also gegenueber dem Scalarmode einen Faktor von 6.1.

Der Faktor gegenueber der 360/91 muss zu deren Gunsten korrigiert werden, da durch konsequente (aber muehsame) Ausnutzung des "Loopmodes" der 91 mit FORTRAN H ca. 20-50% Leistung gewonnen werden koennen.

### 3. Folgerungen

Die Ergebnisse der Benchmarks demonstrieren eindeutig die Leistungsfähigkeit der CRAY-1:

- (1) Die Umstellung der Programme auf den CRAY-1 Computer ist einfach zu bewerkstelligen.
- (2) Die Compilationszeiten (Appendix 1) sind so kurz (Faktor 20-30 gegenüber IPP-Rechnern), dass selbst für grosse Programme keine Object/Load-Library Organisation nötig scheint.
- (3) Die Ergebnisse der Zeitmessungen von Testkernen (Floating Point) bestätigen im wesentlichen die theoretischen Herstellerangaben über die Schnelligkeit der CRAY-1. (Faktor 10 gegenüber der 91, Faktor 25 gegenüber der 470V/6).
- (4) Die Zeitmessungen für realistischere Kerne (2/3 vektorisiert) ergaben für Vektoren mittlerer Länge (72) einen Faktor 10 gegenüber der 91 und einen Faktor 18 gegenüber der 470V/6 (Appendix 2).
- (5) Das wichtigste Ergebnis ist der Leistungsvergleich an Hand grosser MHD Programme:

Die im wesentlichen ungeänderten Codes laufen auf der CRAY-1 etwa 5-10 mal schneller als auf den beiden IPP-Rechnern (Appendix 3).

Relativ geringfügige Änderungen der Programme (um die Vektorisierung von weiteren inneren Schleifen zu ermöglichen) ergaben eine weitere Leistungssteigerung um einen Faktor 2.

- (6) Das Fazit, dass die CRAY-1 einen realistischen Leistungsfaktor von 10 gegenüber der IBM 360/91 und einen Faktor 20 gegenüber der Amdahl 470V/6 erbringt (für Programme mit starker Gleitkommabenutzung in 64 bit Genauigkeit), wird durch die extremen Leistungsfaktoren von 25 bzw. 40 für einzelne Routinen noch unterstrichen.

Die Erweiterung der Rechenkapazität des IPP um eine Größenordnung würde nicht nur eine verbesserte Unterstützung der Experimente durch numerische Rechnungen ermöglichen, sondern sie würde es insbesondere erlauben, ganz neue Probleme anzugehen, die auf Rechnern bisheriger Leistungsklassen nicht gelöst werden konnten.

Appendix 1: Leistungsvergleich: Compiler

CRAY-1 FORTRAN Compiler  
 IBM 360/91 FORTRAN-H extended  
 AMDAHL 470V/6 FORTRAN-H extended

| Programm | L [CRAY-Comp.] | L [CRAY-Comp.] |
|----------|----------------|----------------|
|          | L [Fo-H (91)]  | L [Fo-H (470)] |
| BRM      | 18.2           | 9.9            |
| UHS      | 29.9           | 16.5           |
| ROC      | 33.9           | 16.3           |
| HEW      | 35.4           | 17.9           |
| MCK      | 35.6           | 22.4           |



Appendix 2: Leistungsvergleich: MHD-Code Schleifen

Ausfuehrung typischer Schleifen aus 3D MHD-Codes

| Vektorlaenge | L [CRAY-1] | L [CRAY-1] |
|--------------|------------|------------|
|              | L [360/91] | L [470V/6] |
| REAL*4       |            |            |
| 12           | 6.19       | 6.84       |
| 72           | 10.59      | 10.74      |
| REAL*8       |            |            |
| 12           | 5.98       | 12.33      |
| 72           | 11.11      | 17.93      |

Appendix 3: Leistungsvergleich: MHD-Code Programme

Ausfuehrung grosser MHD-Codes:

| Programm       |            | L [CRAY-1] | L [CRAY-1] |
|----------------|------------|------------|------------|
|                |            | L [360/91] | L [470V/6] |
| BRM            | 2D-CODE    |            |            |
| REAL*8         | (64x64)    | 5.7        | 8.4        |
| UHS            | 3D-CODE    |            |            |
| REAL*8         | (12x32x16) | 6.8        | 10.0       |
| ROC            | 3D-CODE    |            |            |
| REAL*8         | ( 5x 5x 5) | 4.0        | 8.1        |
| REAL*8         | (10x10x10) | 5.1        | 9.5        |
| REAL*8         | (20x20x10) | 5.6        | 10.3       |
| HEW            | 3D-CODE    |            |            |
| REAL*4         |            | ---        | 2.8        |
| REAL*8         |            | ---        | 3.9        |
| MCK            |            |            |            |
| REAL*4         |            | 9.6        | 5.3        |
| REAL*8         |            | 9.6        | 7.0        |
| BRM: optimiert |            | 9.8        | 14.5       |
| UHS: optimiert |            | 12.4       | 18.7       |