

・最適化による計算実行時間の短縮

50元マトリックスを作成し、その逆行列の計算および単純加減乗除算を行なうプログラムを例として、計算時間の比較を行なう。

	B \bar{O} S \cdot F \bar{O} RTRAN	B \bar{O} S-2 \cdot F \bar{O} RTRAN
マトリックス作成時間(S I N関数を使用)	3.4 秒	1.5 秒
逆行列計算時間	38.2 秒	33.2 秒
単純加減乗除演算時間	2.0 秒	0.9 秒
結果の印刷時間(L Pより)	0.7 秒	1.1 秒
プログラム読み込みから計算終了まで	1分32 秒	1分53 秒

以上のように計算実行時間についてみると、B \bar{O} S-2 \cdot F \bar{O} RTRANが相当速くなっているが、総計時間についてはB \bar{O} S-2の方が20秒近く遅くなっている。ところが、EDIT I \bar{O} N-8のB \bar{O} S-2 \cdot F \bar{O} RTRANには最適化機能があり、配列要素の添字計算に対して計算実行時間をもっと速めることが出来る。この機能を利用するには、カタログカード□の次に

／PARA□ \bar{O} PT=1

なるカードを入れればよい。

先程のプログラムにこのカードを入れて実行すると

マトリックス作成	1.4 秒
逆行列計算	14.4 秒
単純加減乗除	0.9 秒
結果印刷	1.1 秒
総時間	1分34 秒

となり、計算量の多いプログラムに対してはB \bar{O} S-2 \cdot F \bar{O} RTRANの方が有利となることがわかる。

2 \cdot B \bar{O} S-2 \cdot F \bar{O} RTRANに新しい機能が加わった。

1) デバック機能(文法編 P129, 手引書 P99参照)

プログラムの実行経過を調べると次のような機能をもつ。

- ・SUBCHK……配列宣言で確保した領域を越えてその配列を使用しているかどうかをチェックする。
- ・TRACE……指定された特定のステートメントナンバーのステートメントが実行されたことを表示する。
- ・SUBTRACE……実行がサブルーチンへ移るごとに、そのサブルーチン名を表示する。
- ・INIT……ある特定の変数の値が変わるごとに、その値を印刷する。
- ・DISPLAY……ある特定の变数や配列の値を任意の時点で表示する。

2) ERR \bar{O} R

通常、計算実行中に起きる軽度なエラーは適当に修正されて計算が続行される。B \bar{O} S

・F \bar{O} RTRANにおいてよく経験される。

ERR \bar{O} R 5054 (C \bar{O} NTINUE)

ERR \bar{O} R 5055 (C \bar{O} NTINUE)

⋮

等が、その例であるがこの場合どの演算中に起ったエラーかを見つけるのが困難であり、

またこういう表示が何ページにもわたって打ち出されることがある。
 B O S - 2 · F O R T R A Nにおいて、この種のエラーは親切にもコメントつきで表示される。またこの種のエラーの回数が指定でき、その回数を越えると計算を打ち切るようにすることができる。この回数を指定するには、プログラムの適当な位置に

CALL ERR LMT (I 1, I 2)

なるステートメントを入れる。(手引書 P 108参照)

ここで I 1 ……エラー番号 3 0 0 0 ~ 3 0 0 8 を除くエラーに対し、その打ち切り回数を指定する。

I 2 ……エラー番号 3 0 0 0 ~ 3 0 0 8 のエラーに対し、その打ち切り回数を指定する。

3) リスト・ダイレクト入出力 (手引書 P 113参照)

通常の入出力文 (R E A D 文や W R I T E 文) は、 F O R M A T ステートメントを必要とするが、リストダイレクト入出力文においては F O R M A T を必要としない。この入出力文は通常の手式付入出力文で F O R M A T の文番号を記述する所に・を記入する。カードよりデータを読むときに、しばしば F O R M A T の記述をまちがうことがあるが、このステートメントによりかなり自由にデータを読み込むことができる。例として次のようなデータカードを

データカード① $\sqrt{\square 25 \square 18, \square, 1986, 34}$

データカード② $\sqrt{2 \cdot 25.0 \square 18.0 \square \nabla A B C D \nabla}$

R E A D (5, *) I I, J J, X X, Y Y

なるステートメントで読み込むと、データカード①を読み込んだとき

I I = 25, J J = 18, X X = 不変, Y Y = 1986, 34

データカード②を読み込んだとき

I I = 25, J J = 25, X X = 18.0, Y Y = $\nabla A B C D \nabla$

となる。同じ値のデータ (例えば 25) を n 個読み込ませるにはデータカードに n · 25 と穿孔すればよい。

4) F O R M A T 文

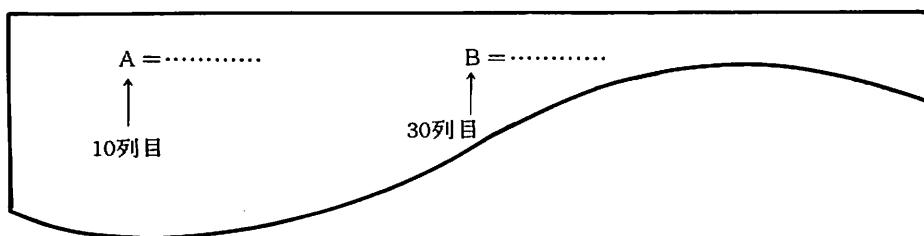
- ・ F O R M A T 文の文字欄記述子として、 ∇ …… ∇ 型記述子が使用できる。
- ・ 位置設定記述子 (T n)

ラインプリンタへ出力する際に、左から n 列目に出力したい時、 T n とする。例えば、

W R I T E (6, 5) B, A

5 \square F O R M A T (1 H \square , T 30, $\nabla B = \nabla$, F 7.2, T 10, $\nabla A = \nabla$, F 7.2)

の時、結果は次のようになる。



5) 倍精度複素数演算が可能

IMPLICIT COMPLEX * 16 (A, B)

なるステートメントによって、A, Bで始まる名前をもつ変数はすべて倍精度複素数となる。倍精度複素数どうしの加減乗除および倍精度複素数を底とし、単精度整数を指数とするべき乗演算が可能である。

- 参考文献
- ・ B O S - 2 ・ F O R T R A N 使用手引書
(3 9 0 P 0 6 4 1 - 1) ……文中手引書
 - ・ B O S - 2 ・ F O R T R A N 文法編
(S P 0 6 1 - 3 - 3) ……文中文法書

1 / 0 チャンネル

例年、今頃から翌年1～2月にかけてセンターに出入りする人がしだいに増え出します。

外の風が冷たいせいか両手をポケットに入れてかけこんでくる人、何か良いことがあったのかニコニコと入ってくる人、もう計算が済んでいるころだろうとTELで確かめもせず、いさぎよくセンターへかけこんでみたものの、さにあらず、せっかく急いできたの……これではお腹の虫がおさまらないとばかりに受付に尋ね、ショボショボと帰る人、センターはたゞ単に処理さえすれば良いのに一体何をしているのかと言わんばかりの人、等々…… e t c

毎年、ユーザはかわってもかわらないのは計算依頼の山です。混雑してくるといつも何故もっと早くからやらないのか、いったい全体こんなに多くのジョブを処理する必要があるものなのか（こんなことを書くとユーザに叱られますが。）とぐちりたくなります。これから日を増すごとに結果待ちのユーザにとってはイライラが解消されないと思いますが、そこは持ち前の若さでがんばって下さい。

(センター員より)

