

UNIVERSIDADE FEDERAL DA PARAÍBA
PRO-REITORIA PARA ASSUNTOS DO INTERIOR
CENTRO DE CIÊNCIAS E TECNOLOGIA

RELATÓRIO FINAL

ESTAGIÁRIA: JANEIDE PEREIRA DE ALBUQUERQUE

ORIENTADOR: LIKISO HATTORI

Campina Grande, 14 de Julho de 1979



Biblioteca Setorial do CDSA. Março de 2021.

Sumé - PB

ÍNDICE

I Apresentação

II Introdução

III Apresentação das Etapas de Trabalho

A) ESTUDOS NECESSÁRIOS

1. JCL (Job Control Language)

1.1 - Porque Necessitamos JCL

1.2 - O que é JCL

2. Fundamentos de OS/VS1

2.1 - O que é um Sistema Operacional

2.2 - Funções do Programa de Controle do OS/VS1

2.3 - Programas de Processamento de OS/VS1

2.4 - Características e Facilidades do OS/VS1

3. Utilitários

4. Manipulação de Bibliotecas

4.1 - O que é uma Biblioteca de Programas

B) IMPLEMENTAÇÃO DA VERSÃO TRADUZIDA DO STUNT DA BIBLIOTECA WATFIV.WATLIB

1. Descrição da Biblioteca WATFIV.WATLIB

2. O Pacote de Rotinas do STUNT

2.1 - Descrição

2.2 - Implementação

IV Conclusão

V Recursos Utilizados

VI Anexos

I. APRESENTAÇÃO

O presente Relatório Final narra as atividades do Estágio Supervisionado, (que conforme a RESOLUÇÃO 02/75 prevista na Portaria 159-MEC de 14 de julho de 1975, é exigido para integralização dos créditos), da aluna JANEIDE PEREIRA DE ALBUQUERQUE, matrícula 7711176-0, do Curso de Formação de Tecnólogos em Processamento de Dados, do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba.

O referido Estágio Supervisionado foi realizado no Departamento de Sistemas e Computação do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba em Campina Grande, sob a orientação do Professor LIKISO HATTORI.

O citado Relatório Final é destinado à apreciação da Coordenação do Curso de Formação de Tecnólogos a nível Superior em Processamento de Dados.

II. INTRODUÇÃO

O nosso Estágio Supervisionado consta, basicamente, de duas fases que podemos dividir da seguinte forma: Primeira fase engloba os estudos e pesquisas sobre Linguagem de Controle de Jobs, Fundamentos de OS/VSl, Utilitários, e Manipulação de Bibliotecas, que são necessários para o cumprimento da segunda fase que é a Implementação na Biblioteca de Programas WATFIV.WATLIB, de uma versão traduzida por nós do pacote de rotinas recebido da Universidade de Toronto, Canadá.

O referido pacote do STUNT (Software for Teaching Undergraduates Numerical Techniques) contém rotinas para Solução de Equações não Lineares, Algebra Linear, Interpolação e Aproximação, Quadratura e Equações Diferenciais Ordinárias.

III. APRESENTAÇÃO DAS ETAPAS DE TRABALHO

A) ESTUDOS NECESSÁRIOS

Levando em consideração a necessidade de utilizar JCL (Job Control Language), alguns tipos de UTILITÁRIOS, manipular com biblioteca de programas, e que, no decorrer do nosso Curso tivemos sô noções elementares dos citados assuntos foi preciso fazer um estudo sobre os mesmos, além de noções do Sistema Operacional OS/VSl do IBM/370-145.

1. Linguagem de Controle de Jobs (JCL)

1.1 - Porque Necessitamos JCL

Uma grande variedade de tarefas podem ser executadas pelo Sistema Computador. Este Sistema requer uma série de informações que determine quais tarefas o programador deseja que ele execute e onde os dados para as mesmas estão armazenados; e para isto usamos uma linguagem de programação especial que é o JCL.

1.2 - O que é JCL

A Linguagem de Controle de Jobs (JCL), como muitas outras linguagens de programação, tem várias

diferentes características; entretanto, existem somente três formas básicas de cartões de declaração de JCL: Cartões de Comentário, que servem para propósitos de documentação; Cartões de Controle do Sistema, através dos quais o programador diz ao computador quem é o usuário, quais tarefas devem ser executadas e onde os dados devem entrar e/ou sair; existem três tipos de cartão de comando que são: Cartão JOB, Cartão EXEC e Cartão DD, os quais são compostos de vários parâmetros; e finalmente temos os Cartões Delimitadores que servem para indicar o fim de um Job ou de um conjunto de dados.

2. Fundamentos de OS/VSl (Operating System Virtual Storage 1).

2.1 - Que é um Sistema Operacional.

Um Sistema Operacional é uma coleção de programas, escritos e supridos pelos fabricantes, designados para fornecer a eficácia total na operação de uma instalação de processamento de dados. O Sistema Operacional tem a seguinte esquematização:

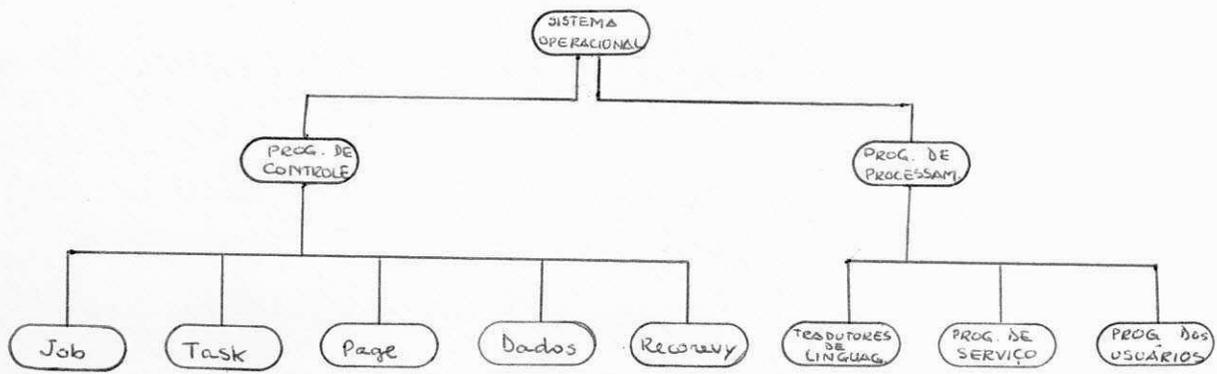


FIGURA 1 - CONFIGURAÇÃO DO SISTEMA OPERACIONAL

2.2 - Função dos Programas de Controle do OS/VSI

As rotinas de Programas de Controle têm cinco principais funções como mostra a figura abaixo:

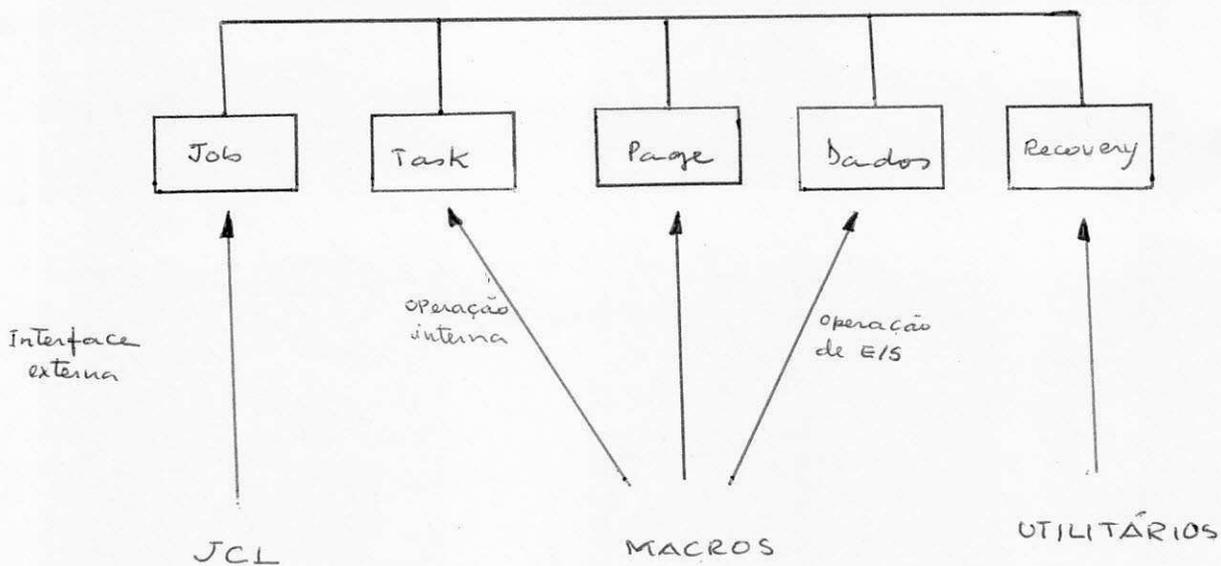


FIGURA 2 - INTERFACE



1. Gerenciamento de Jobs - Analisa o input Stream; aloca os dispositivos de E/S; seleciona Jobs para execução; lê e imprime os dados do Job e fornece comunicação entre o operador e o sistema;
2. Gerenciamento de Tarefas é também referenciado como Supervisor. Suas funções são Supervisão de: Interrupção, Tarefa, Memória Virtual, Conteúdos, Paginação, Overlay e Tempo;
3. Gerenciamento de Recuperação (Recovery Management) - coleta e grava status de erros, faz a recuperação de função, do sistema executa restart do suporte do sistema e faz os reparos do sistema;
4. Gerenciamento de Dados - suas rotinas são: Supervisão de E/S; os Métodos de Acesso, Gerenciamento de Catalogos, DADSM e Open/Close/end-of-volume (EOV);
5. Gerenciamento de Paginação - rotinas adicionais compostas de Supervisor de memória real, executa armazenamento para paginação e manutenção da performance do sistema e suporte de Virtual = Real.

2.3 - Programas de Processamento do OS/VSl

Como programas de processamento de um Sistema Operacional OS/VSl temos:

1. Tradutores de Linguagem - compiladores e o assembler do sistema;
2. Programas de Serviço - Linkage/Editor, Loader;
3. Programas do Sistema escritos pelo usuário.

2.4 - Características e Facilidades do OS/VS1

- Inicialização automatizada;
- Memória Virtual;
- Extensa Capacidade de Multiprogramação;
- Scheduling Independente de Jobs;
- Facilidades de Execução Virtual = Real;
- Facilidade de Gerenciamento de Sistema (SMF);
- Tempo de CPU de um Step de Job;
- Limite de Tempo de "Wait";
- Checkpoint/Restart;
- Recovery Management Suporte;
- Redefinição do tamanho de Partição;
- Systems Input Readers;
- Input Stream from Disk;
- Dynamic Dispatching;
- Systems Output Writers;
- Direct System Output Writers;
- Reinício do Sistema sem Perda dos Jobs enfileirados;

- Execução Concorrente de Tarefas dentro de uma Partição;
- Remote Entry Services (RES).

3. Utilitários

O Sistema Operacional do sistema /370 fornece programas utilitários para organização e manutenção de dados. Existem três classes de programas utilitários determinadas pela função que o programa executa e pela maneira que o programa é controlado.

- Programas Utilitários do Sistema, que são usados para manutenção dos dados de controle do sistema em um nível organizacional ou de sistema. Como exemplo, podemos ter:

IEHDASOR - para inicializar volumes de acesso direto ou rearmazenar dados;

IEHIOSUP - para atualizar entradas no supervisor das bibliotecas;

IEHLIST - para listar dados de controle do sistema, conforme mostra o anexo 1;

IEHPROGRAM - para construir e manter dados de controle do sistema.

- Programas Utilitários de Arquivos, que são usados para reorganizar, trocar ou comparar dados a nível de arquivos e/ou registros, tais como:

IEBCOPY - para copiar, comprimir ou intercalar arquivos, para selecionar ou excluir determinados membros numa operação de cópia;

IEBGENER - para copiar registros de um arquivo sequencial;

IEBTPCH - para imprimir ou perfurar registros de um arquivo sequencial ou particionado, conforme anexo 3;

IEBUPDTE - para fazer trocas em um arquivo sequencial ou particionado, como mostrado no anexo 5.

- Programas Utilitários Independentes, que são usados para preparar dispositivos para uso do sistema quando o Sistema Operacional não está disponível, como exemplo citamos:

IBCDASDI - usado para inicializar volumes de acesso direto, e assinalar trilhas alternativas.

4. Manipulação de Bibliotecas

4.1 - O que é uma Biblioteca de Programas

Uma Biblioteca de Programas é uma coleção organizada de programas, rotinas e/ou subrotinas (que servem de padrão por terem sido testados), com suficiente documentação para permitir o uso dos mesmos por pessoas diferentes dos respectivos autores.

As Bibliotecas de Programas podem ser:

- Do Sistema - são arquivos particionados que contém programas de controle para o sistema, como exemplo temos: o SYS1.LINKLIB cujo conteúdo são os compiladores de linguagem, o SYS1.SYSJOBQE onde estão armazenados os streams e filas de E/S, o SYS1.LOGREC, que contém dados estatísticos sobre os erros da máquina, o SYS1.PROCLIB que contém e as procedures catalogadas, entre outros não citados. Grande parte destes arquivos é armazenada no módulo de carga (código binário);

- Do Usuário - são arquivos particionados que são criados para vários propósitos, como exemplo temos a SYS2.LINKLIB que é usada pela ATECEL e NPD para catalogar seus programas, e são armazenados em módulo de carga (código binário), dependendo da linguagem utilizada; a WATFIV.WATLIB que contém rotinas numéricas para utilização dos alunos e está armazenado sob a forma de programa fonte.

As Bibliotecas de Programas são armazenadas na organização de arquivos particionada.

Estes arquivos particionados são divididos em Membros que estão dispostos sequencialmente sob o comando de um Diretório, conforme esquematizamos a seguir.

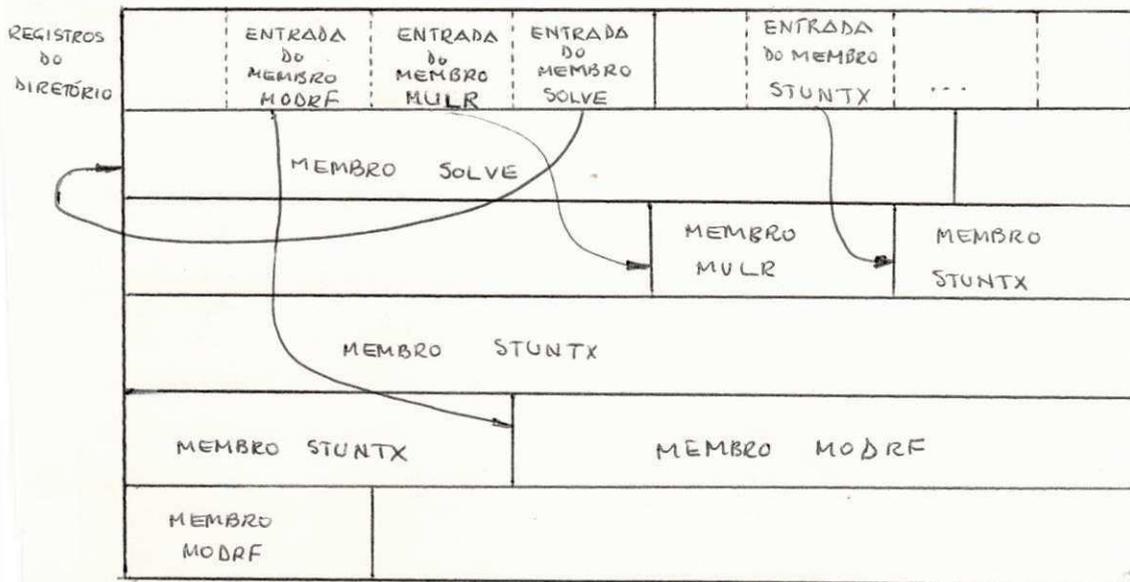


FIGURA 3 - EXEMPLO DA BIBLIOTECA WATFIV.WATLIB



FIGURA 4 - EXEMPLO DO BLOCO DO DIRETÓRIO

O Diretório de um arquivo particionado localiza-se no início da área alocada para o arquivo de um volume de acesso direto.

Quando deletamos um Membro, o nome deste é removido do Diretório, mas o espaço usado por este Membro não pode ser reusado até que o arquivo seja reorganizado.

B) IMPLEMENTAÇÃO DA VERSÃO TRADUZIDA DO STUNT NA BIBLIOTECA DE PROGRAMAS WATFIV.WATLIB

1. Descrição da Biblioteca WATFIV.WATLIB

A Biblioteca WATFIV.WATLIB é um arquivo particionado que está armazenado em forma de fonte. Uma das desvantagens de armazenarmos uma biblioteca desta forma é que, ao invocarmos um programa, além da link edição o mesmo ainda deve ser compilado, enquanto que se o arquivo fosse armazenado sob a forma de programa objeto, só seria necessário a link edição; e uma das vantagens de ser armazenada sob a forma de programa fonte, é que cada rotina tem documentação interna orientando como deve ser usado cada parâmetro de entrada/saída, facilitando assim, sua utilização pelos seus usuários que em geral são alunos de graduação e pós-graduação do Centro de Ciências e Tecnologia.

2. O Pacote de Rotinas do STUNT

2.1 - Descrição

O Pacote de Rotinas do STUNT contém rotinas para Álgebra Linear, Soluções de Equações não Lineares, Interpolação e Aproximação, Quadratura e Equações Diferenciais Ordinárias. O referido pacote foi recebido da Universidade de Toronto, Canadá, e suas rotinas devem ser invocadas por programas escritos em FORTRAN e/ou compilados pelo compilador WATFIV.

A maioria das rotinas estão disponíveis em precisão simples e dupla. As de precisão dupla são precedida lebra D.

O referido pacote de rotinas do STUNT estava todo documentado em inglês, o que desfavorecia os alunos que necessitavam utiliza-las, e não tinham conhecimento do idioma, esta foi a razão pela qual o grupo de Análise Numérica do Departamento de Sistemas e Computação resolveu implementar a versão traduzida das rotinas.

2.2 - Implementação

As etapas de implementação propriamente dita foram realizadas conforme a seguinte descrição:

Primeiramente utilizamos o utilitário IEHLIST para listar o diretório da Biblioteca de Programas WATFIV.WATLIB, como mostrado no anexo 1;

Uma vez listado todo o diretório, solicitamos para cada rotina a perfuração de cada membro através do utilitário IEBTPCH como mostrado no anexo 3;

Mandamos listar cada membro da seguinte maneira, antes e depois da implantação

```
$JOB          JANY,LIBLIST
              CALL ROTINA
              STOP
              END
$ENTRY
```

Depois de listado fizemos a tradução de cada membro e testamos a sua sintaxe;

Uma vez traduzido fizemos os testes de sintaxe e implementamo-lo na referida Biblioteca, através do utilitário IEBUPDTE de acordo com o anexo 4;

Após a implementação da rotina no arquivo testamo-la outra vez, conforme mostra o anexo 5.

IV. CONCLUSÃO

A implementação da versão traduzida das rotinas do STUNT ainda não chegou a sua fase final, uma vez que a execução das etapas de trabalho não depende só da Estagiária, e sim também de outros parâmetros, tais como disponibilidade de nosso orientador que está muito sobrecarregado de atividades, tempo de rodadas dos programas no NPD-CG, etc; mas nos comprometemos a terminar a referida implementação tão logo quanto possível.

No que diz respeito às atividades desempenhadas, muito valeu a experiência, uma vez que, tivemos oportunidade de aprender matérias que não foram ensinadas no decorrer do Curso de Processamento de Dados, como é o caso de Fundamentos de OS/VSl, Manipulação de Bibliotecas ou daquelas que só foi dada uma introdução muito elementar como o JCL (Linguagem de Controle de Jobs) e utilização de Utilitários.


JANILDE PEREIRA DE ALBUQUERQUE
Estagiária


LIKISO HATTORI
Orientador

V. RECURSOS UTILIZADOS

A) Bibliografia

1. IBM - System Reference Library
Os Data Management Service Guide
2. OS/VS1 - Os MFT + DATA FEATURE + JES - S/370.
3. IBM OS/VS1 Planning and User Guide.
4. IBM OS/VS1 JCL REFERENCE
5. STUNT - Software for Teaching Undergraduates Nu
merical Techniques.
R.L. Johnston and C. Addison.
6. IBM System/370 Operating System Utilities
Systems Reference Library

B) O Sistema 370/145 Instalado no Núcleo de Processamen
to de Dados de Campina Grande.

VI . ANEXOS

\$JOB JANEIDE,LIBLIST

1 CALL DCOMP

2 STOP

3 END

\$ENTRY

SUBROUTINE DCOMP (NDIM,N,A,NPIV,IND)

00000

C*****00000

C *00000

C ESTA SUBROTINA, DECOMPOSICAO COM PIVOTAMENTO PARCIAL, EXECUTA A *00000

C ELIMINACAO DE GAUSS OU, EQUIVALENTEMENTE UMA FATORACAO DA *00000

C MATRIZ N*M ARMAZENADA EM A. *00000

C AO TERMINO, A CONTERA A MATRIZ TRIANGULAR INFERIOR DOS MULTIPLI- *00000

C CADORES USADOS NA ELIMINACAO BEM COMO A MATRIZ TRIANGULAR SUPE- *00000

C RIOR U. RESULTANTE DA ELIMINACAO. *00000

C *00000

C A MATRIZ A EH CONSIDERADA SINGULAR SE ALGUM PIVOT DURANTE A *00000

C ELIMINACAO FOR MENOR QUE UM EPSILON TAI QUE *00000

C 1 + EPSILON = 1. *00000

C SE ISTO ACONTECE DCOMP TERMINA A ELIMINACAO, MAS A MATRIZ TRIAN- *00000

C GULAR SUPERIOR RESULTANTE SERA SINGULAR. *00000

C *00000

C ESTA ROTINA PODERA SER USADA JUNTAMENTE COM A ROTINA SOLVE PARA *00000

C ENCONTRAR A SOLUCAO DE UM SISTEMA DE EQUACOES LINEARES. *00000

C *00000

C-----*00000

C *00000

C CHAMADA CALL DCOMP (NDIM,N,A,NPIV,IND) *00000

C *00000

C PARAMETROS *00000

C *00000

C NDIM -UM NUMERO INTEIRO QUE INDICA O NUMERO DE LINHAS NO ARRAY *00000

C A, CONFORME DECLARADO NO PROGRAMA INVOCADOR. *00000

C N -UMA CONSTANTE INTEIRA QUE DA O TAMANHO DO SISTEMA A SER *00000

C RESOLVIDO. *00000

C A -UM ARRAY DE DIMENSAO 2, TAMANHO NDIM*N, ARMAZENANDO A *00000

C MATRIZ A SER DECOMPOSTA, NO RETORNO O CONTEUDO DE A EH *00000

C SUBSTITUIDO PELOS RESULTADOS DA DECOMPOSICAO LU. *00000

C NPIV -UM VETOR INTEIRO, DE DIMENSAO N, QUE NAO PRECISA SER *00000

C INICIALIZADO PELO PROGRAMA INVOCADOR. ESSE VETOR REGIS- *00000

C TRARA O REARRANJO DAS LINHAS. *00000

C IND -UM INTEIRO QUE INDICA SE A MATRIZ EH OU NAO SINGULAR. *00000

C IND = 0, -A NAO EH SINGULAR *00000

C IND = -1, A EH SINGULAR *00000

C *00000

C-----*00000

C *00000

C EXEMPLO DE UM PROGRAMA ATIVADOR *00000

C DIMENSION A(5,4),NPIV(4) *00000

C READ,N,(A(I,J),J=1,N),I=1,N) *00000

C CALL DCOMP(5,N,A,NPIV,IND) *00000

C IF(IND.NE.0) PRINT,'MATRIZ SINGULAR' *00000

C DO 1 I = 1,N *00000

C IPIVOT = NPIV(2) *00000

C PRINT,(A(IPIVOT,J),J =1,N) *00000

C1 CONTINUE *00000

C PRINT2 *00000

C2 FORMAT('0'.6X,'LINHA ATUAL LINHA ORIGINAL') *00000

C DO 3 I = 1,N *00000

C PRINT,NPIV(I),I *00000

```

03  CONTINUE *0000
C  STOP *0000
C  END *0000
C$ENTRY *0000
C  DADOS *0000
C  *0000
C  N (= DE INCOGNITAS - CONSTANTE) 1. 2. 3. ETC *0000
C  *0000
C  REFERENCIA ELEMENTARY NUMERICAL ANALYSIS *0000
C  AN ALGORITHMIC APPROACH 0000
C  DE CONTE AND DE ROOR *0000
C  *0000
C  TRABALHO DO GRUPO DE ANALISE NUMERICA *0000
C  DEPARTAMENTO DE SISTEMAS E COMPUTACAO *0000
C  UNIVERSIDADE FEDERAL DA PARAIBA *0000
C  *0000
C *****0000
5  DIMENSION A(NDIM,N),NPIV(N) 0000
6  INC = 0 0000
C  0000
C  VERIFICA SE O SISTEMA SO TEM UMA INCOGNITA 0000
C  0000
7  IF(N.EQ.1) RETURN 0000
C  0000
C  INICIALIZA VETOR DE PIVOT 0000
C  0000
8  DO 10 I = 1,N 0000
9  NPIV(I) = I 0000
10 10 CONTINUE 0000
C  0000
C  LOOP PRINCIPAL PARA ELIMINACAO DE GAUSS 0000
C  0000
11 NM1 = N-1 0000
12 DO 22 I = 1,NM1 0000
C  0000
C  PESQUISA COLUNA PARA ENCONTRAR O MAIOR PIVOT. 0000
C  ISTO EH. MAX (A(J,I) , I )= J )= N 0000
C  0000
13 COLMAX = 0. 0000
14 DO 14 J = I,N 0000
15 HOLD = ABS(A(NPIV(J),I)) 0000
16 IF(HOLD.GE.COLMAX) GO TO 14 0000
17 COLMAX = HOLD 0000
18 NROW = J 0000
9 14 CONTINUE 0000
C  VERIFICA SE A EH SINGULAR. A MATRIZ EH CONSIDERADA 0000
C  SINGULAR SE COLMAX (O VALOR ABSOLUTO DO PIVOT) EH 0000
C  EQUIVALENTE A ZERO. ISTO EH. 0000
C  1.0 + COLMAX = 1.0 0000
C  SE ISSO ACONTECE A ROTINA PROSEGUE NO (I+1)-ESIMO 0000
C  ESTAGIO DA ELIMINACAO. 0000
C  0000
20 IF(1. + COLMAX.NE.1) GO TO 16 0000
21 IND = -1 0000
22 GO TO 22 0000
C  0000
C  SE UMA TROCA FOR NECESSARIA, ALTERA O VETOR NPIV. 0000
C  0000
23 16 IPIVOT = NPIV(NROW) 0000
24 IF(NROW.EQ.I) GO TO 17 0000

```

```

25          NPIV(NROW) = NPIV(I)          00001150
26          NPIV(I) = IPIVOT             00001160
C
C      OS MULTIPLICADORES PARA COMPUTACAO DAS LINHAS.
C      RESTANTES SAO DETERMINADOS E A ELIMINACAO EH
C      EXECUTADA. O VALOR DE CADA MULTIPLICADOR EH
C      ARMAZENADO NA POSICAO DO ELEMENTO ELIMINADO.
C
27 17          IPI = I + 1                00001230
28          DO 20 J = IPI,N              00001240
29          JPIVOT = NPIV(J)             00001250
30          AMULT = A(JPIVOT,I)/A(IPIVOT,I) 00001260
31          A(JPIVOT,I) = AMULT          00001270
32          DO 18 K = IPI,N              00001280
33          A(JPIVOT,K)=A(JPIVOT,K)-AMULT*A(IPIVOT,K) 00001290
34 18          CONTINUE                   00001300
35 20          CONTINUE                   00001310
36 22          CONTINUE                   00001320
37          IF(1.+ABS(A(NPIV(N),N)). EC . 1.) IND = -1 00001330
38          RETURN                        00001340
39          END                            00001350

```

ERROR INVALID TYPE OF ARGUMENT IN REFERENCE TO SUBPROGRAM DCOF

PROGRAM WAS EXECUTING LINE 1 IN ROUTINE M/PROG WHEN TERMINATION OCCURRED

CORE USAGE OBJECT CODE= 1616 BYTES,ARRAY AREA= 0 BYTE,TOTAL AREA AVAILABLE= 170984 BYTES

DIAGNOSTICS NUMBER OF ERRORS= 1, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSIONS= 0

COMPILE TIME= 0.74 SEC,EXECUTION TIME= 0.01 SEC. WATFIV - JI 1973 V114 12.48.30 WEDNESDAY 6 JUN 79

SYSIN

NEW MASTER

IERUPDTE LOG PAGE 0001

./ ADD NAME=NEWTN.LEVEL=00.SOURCE=0.LIST=ALL
./ NUMBER NEW1=10.INCR=10

C SUBROUTINE NEWTN (F,FPRIME,XO,XTOL,FTOL,ITMAX,IER,LISTA) 00000010
C *****00000020
C *00000030
C ESTA SUBROTINA UTILIZA O METODO DE NEWTON PARA OBTEN- *00000040
C MAO PARA UMA RAIZ DE UMA FUNCAO ESPECIFICA. O USUARIO DEVE *00000050
C FORNECER UMA APROXIMACAO INICIAL X(0). A ROTINA CALCULA SUCE- *00000060
C SIVAMENTE APROXIMACOES X(I) ATÉ QUE UMA DAS SEGUINTE- *00000070
C DE TERMINDO SEJA SATISFEITA *00000080
C 1. MODULO DE F(X(I)) MENOR OU IGUAL A FTOL E UMA DAS CONDIC- *00000090
C MODULO DE (X(I)-X(I-1)) MENOR OU IGUAL A XTOL OU *00000100
C MODULO DE X(I) MENOR OU IGUAL A XTOL**2 *00000110
C 2. ITMAX ITERACOES FORAM EXECUTADAS. *00000120
C *00000130
C ----- *00000140
C *00000150
C CHAMADA CALL NEWTN (F,FPRIME,XO,XTOL,FTOL,ITMAX,IER,LISTA) *00000160
C *00000170
C PARAMETROS *00000180
C *00000190
C F -SUBPROGRAMA FORNECIDO PELO USUARIO PARA AVALIAR A FUN- *00000200
C CAO DE INTERESSE. ELE DEVE TER A FORMA *00000210
C FUNCTION F(X) *00000220
C REAL X *00000230
C . *00000240
C . *00000250
C . *00000260
C FPRIME -SUBPROGRAMA FORNECIDO PELO USUARIO PARA AVALIAR A DERI- *00000270
C VADA DA FUNCAO EM QUESTAO. DEVE TER A MESMA FORMA DE F. *00000280
C XO -UMA VARIÁVEL REAL CUJO VALOR É A APROXIMACAO INICIAL *00000290
C DA RAIZ. DEVE SER PASSADA PARA A SUBROTINA COMO VARIÁ- *00000300
C VEL E NAO COMO CONSTANTE. NO RETORNO ESTA VARIÁVEL *00000310
C CONTEM A MAIS RECENTE APROXIMACAO DA RAIZ. *00000320
C XTOL -UMA VARIÁVEL REAL QUE ESPECIFICA A TOLFRANCIA ACEITA- *00000330
C VEL NA MUDANCA RELATIVA *00000340
C MODULO DE (X(I)-X(I-1)) / MODULO DE X(I) *00000350
C NAS SUCESSIVAS ITERACOES. *00000360
C FTOL -UMA VARIÁVEL REAL QUE ESPECIFICA A TOLFRANCIA ACETAVEL *00000370
C DO F(X(I)). *00000380
C ITMAX -UMA VARIÁVEL INTEIRA CUJO VALOR NA CHAMADA A SUBROTINA *00000390
C ESPECIFICA O NUMERO MAXIMO DE ITERACOES DESEJADO. NO *00000400
C RETORNO ITMAX CONFERA O NUMERO DE ITERACOES EXECUTADAS. *00000410
C IER -UMA VARIÁVEL INTEIRA QUE INDICA O STATUS DA ULTIMA *00000420
C ITERACAO EXECUTADA *00000430
C IER = 1 -UMA RAIZ FOI ENCONTRADA SATISFAZENDO AS *00000440
C TOLFRANCIAS ESPECIFICADAS. *00000450
C = 2 -ITMAX ITERACOES FORAM EXECUTADAS, XTOL FOI *00000460
C SATISFEITA MAS NAO FTOL. *00000470
C = 3 -ITMAX ITERACOES FORAM EXECUTADAS, FTOL *00000480
C FOI SATISFEITA MAS NAO XTOL. *00000490
C = 4 -NAO CONVERGIU EM ITMAX ITERACOES. *00000500

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

C           = 5 -AS ITERACOES FORAM INTERROMPIDAS PORQUE A *000005
C           DERIVADA CHEGOU A ZERO. ( ) 1.F-30). *000005
C   LISTA -UMA VARIAVEL INTEIRA USADA PARA CONTROLAR A SAIDA. *000005
C           LISTA = 0 -NAO HA SAIDA. *000005
C           = K -SAIDA A CADA K ITERACOES. *000005
C           ITMAX -SOMENTE VALORES INICIAIS SERAO *000005
C           IMPRESSOS. *000005
C *000005
C   EXEMPLO DE UM PROGRAMA ATIVADOR *000005
C *000006
C   EXTERNAL F,FPRIME *000006
C   READ,X0,XTOL,FTOL,ITMAX,LISTA *000006
C   CALL NEWTN (F,FPRIME,X0,XTOL,FTOL,ITMAX,IER,LISTA) *000006
C   GO TO(12,14,16,18,20),IER *000006
C 12 PRINT,X0,F(X0) *000006
C   GO TO 6 *000006
C 14 PRINT,X0,'FTOL NAO FOI SATISFEITA' *000006
C   GO TO 6 *000006
C 16 PRINT,X0,'XTOL NAO FOI SATISFEITA' *000006
C   GO TO 6 *000007
C 18 PRINT,'NUMERO INSUFICIENTE DE ITERACOES. *000007
C   * A ULTIMA APROXIMACAO FOI',X0 *000007
C   GO TO 6 *000007
C 20 PRINT,'A DERIVADA SE ANULOU PARA X0 =',X0 *000007
C 6 STOP *000007
C   END *000007
C *000007
C   FUNCTION F(X) )----- FORNECIDO PELO USUARIO *000007
C   F = FUNCAO DESEJADA *000007
C   RETURN *000008
C   END *000008
C *000008
C   FUNCTION FPRIME (X) )----- FORNECIDO PELO USUARIO *000008
C   FPRIME = F'(X) *000008
C   RETURN *000008
C   END *000008
C *000008
C$ENTRY *000008
C *000008
C   DADOS *000009
C *000009
C   REFERENCIA ELEMENTARY NUMERICAL ANALYSIS *000009
C   AN ALGORITHMIC APPROACH *000009
C   DE CONTE AND DE BOOR. *000009
C *000009
C   TRABALHO DO GRUPO DE ANALISE NUMERICA *000009
C   DEPARTAMENTO DE SISTEMAS E COMPUTACAO *000009
C   UNIVERSIDADE FEDERAL DA PARAIBA *000009
C *000009
C *****000010
C   XTOL SQ = XTOL*XTOL 000010
C   IER = 4 000010

```

SYSTN

NEW MASTER

IEBUPDTE LOG PAGE 0003

```

NEXT=LISTA
IF (LISTA.GT. 0) WRITE(6,1)
FX0 = F(X0)
LIMIT=ITMAX + 1
DO 14 I=1,LIMIT
  ITER=I-1
  PRIME=FPRIME(X0)
  IF (ABS(PRIME) .GT. 1.E-30) GO TO 10
  IER = 5
  GO TO 16
10  CONTINUE
  DELTAX=FX0/PRIME
  IF ((LISTA.EQ. 0) .OR. ((ITER .NE. NEXT) .AND. (ITER .NE.
+ 0))) GO TO 12
  WRITE(6,2) IER,X0,FX0,DELTAX,PRIME
  NEXT = ITER + LISTA
12  CONTINUE
  X0=X0-DELTAX
  FX0=F(X0)
  IF (ABS(FX0) .LE. FTOL) IER = 3
  IF (ABS(X0) .LE. XTOL .AND. IER.EQ.3) GO TO 16
  IF ((ABS(DELTAX) .LE. XTOL*ABS(X0)) .AND. (IER .GT. 2))
-   IER = IER - 2
  IF (IER .EQ. 1) GO TO 16
14  CONTINUE
  RETURN
16  ITMAX = ITER
  RETURN
1  FORMAT('-. 'ITERACAO ',10,'X',17X,'F(X)',14X,'DELTAX',13X,'FPRIME'
+ '/')
2  FORMAT(' ',3X,13,8X,4(F,7,5X))
END

```

```

00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340

```

ENDUP

```

IEB16T MEMBER NAME (NEWTN ) FOUND IN NM DIRECTORY. TTR IS NOW ALTERD.
IEB18T HIGHEST CONDITION CODE WAS C000000
IEB19T END OF JOB IEBUPDTE.

```

ANEXO 4

```
//JANFIDE JOB MSGLEVEL=1
// EXEC PGM=IEBUPDTF,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSNAME=WATFIV,WATI TR,UNIT=3330,
// DTSP=SFR,VOI=SFR=UEPVS2
//SYSTEM DD DATA
//SYSTEM DD * GENERATED STMT
IEE236T ALLOC. FOR JANFIDE
IEE237T 152 ALLOCATED TO SYSUT2
IEE142T - STEP WAS EXECUTED - COND CODE 0000
IEE285T WATFIV,WATI TR KEPT
IEE285T VOI SFR NOS=UEPVS2.
IEE298T JANFIDE SYSOUT=A.
```

PRINT/PUNCH DATA SET UTILITY

PAGE 0001

PUNCH MAXFIDS=1, TYPORG=PD, MAXNAME=150
MEMBER NAME=IMPRV
RECORD FIELD=(80)
EOF ON SYSIN
END OF DATA FOR SDS OR MEMBER

```
//JANFIDE JOB MSGLEVEL=1
// EXEC PGM=IEBPTCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=JANFIV.WATLIB.DISP=OLD.VOL=SER=UEPVS2.
// UNIT=SYSDA
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
//SYSIN DD * GENERATED STMT
IEF236I ALLOC. FOR JANFIDE
IEF237I 152 ALLOCATED TO SYSUT1
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I WATFIV.WATLIB KEPT
IEF285I VOL SER NOS=UEPVS2.
IEF298I JANFIDE SYSOUT=A.
```

PRINT/PUNCH DATA SET UTILITY

PAGE 0001

PRINT MAXFIELDS=1.TYPORG=PD.MAXNAME=207

MEMBER NAME=MILLR

RECORD FIELD=(80)

EOF ON SYSTM

END OF DATA FOR SDS OR MEMBER


```

MEMBER NAME      MUIR
C *
C * ONCE THE NEW APPROXIMATION HAS BEEN FOUND, THE X AND Y VALUES ARE
C * UPDATED, SO THAT X3 CONTAINS THE LATEST APPROXIMATION, X2 THE
C * PREVIOUS AND X1 THE APPROXIMATION BEFORE THAT.
C *
C *****
C
      X1 = X2
      Y1 = Y2
      X2 = X3
      Y2 = Y3
      Y3 = F(X)
      X3 = X
      IF (I.NE. NEXT) GO TO 10
      NEXT = I + 1 LIST
      WRITE(6,2) I,X3,Y3
10    IF ( CABS(Y3).LE.E101) IER = 3
      IF ( CABS(X3).LE.XTOLSQ .AND. IER.EQ.3 ) GO TO 110
      IF ( CABS(X3-X2) .LE. XTOL*CABS(X3) .AND. IER .GT. 2)
-      IER = IER - 2
      IF ( IER .EQ. 1) GO TO 110
100   CONTINUE
      RETURN
110   ITMAX = I
      RETURN
1    FORMAT(10,' ITERATION',13X,' APPROXIMATION',27X'FUNCTION VALUE' //
+ ' ',1,' INITIAL',5X,2(2(D15.8 ,3X),6X))
2    FORMAT(' ',14,8X,2(2(D15.8 ,3X),6X))
      END

```

```

*****
//JANFIDE      JOB      MSGLEVEL=1
//              EXEC      PGM=IERPTPCH
//SYSPRINT     DD      SYSOUT=A
//SYSUT1       DD      DSNNAME=WATFIV.WATLIB.DISP=OLD.VOL=SER=UEPVS2.
// UNIT=SYSDA
//SYSUT2       DD      SYSOUT=B
//SYSIN        DD      *
//SYSIN DD * GENERATED STMT
IEF236I ALLOC. FOR JANFIDE
IEF237I 152 ALLOCATED TO SYSUT1
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I WATFIV.WATLIB KEPT
IEF285I VOL SER NOS=UEPVS2.
IEF298I JANFIDE SYSOUT=A.B.

```

\$.JOP .JANU.11BI1ST

1 CALL DCOMP
2 STOP
3 END

4 \$ENTRY

SUBROUTINE DCOMP(NDIM,N,A,NPIV,IND)

0000

C
C
C 0000
C 0000

C THIS SUBROUTINE, DECOMPOSITION WITH PARTIAL PIVOTING, DOES
C GAUSSIAN ELIMINATION OR, EQUIVALENTLY, A TRIANGULAR (LU) FACTORIZA-
C TION OF THE N*N MATRIX STORED IN THE ARRAY A. AT COMPLETION, THE
C ARRAY A WILL CONTAIN THE LOWER TRIANGULAR MATRIX OF MULTIPLIERS USED
C IN THE ELIMINATION AS WELL AS THE UPPER TRIANGULAR MATRIX U, THE
C RESULT OF THE ELIMINATION.

C
C THE MATRIX IS ASSUMED TO BE SINGULAR IF SOME PIVOT DURING ELIMINA-
C TION IS SMALLER THAN UNIT ROUND-OFF. IF THIS HOLDS, DCOMP WILL
C COMPLETE THE ELIMINATION BUT THE RESULTING UPPER TRIANGULAR MATRIX
C WILL BE SINGULAR.

C
C THIS ROUTINE CAN BE USED IN CONJUNCTION WITH THE ROUTINE SOLVE TO
C FIND THE SOLUTION TO A SYSTEM OF LINEAR EQUATIONS.

C
C
C 0000

C CODED JANUARY, 1976.
C DEPARTMENT OF COMPUTER SCIENCE,
C UNIVERSITY OF TORONTO.

C
C
C 0000

C CALLING SEQUENCE CALL DCOMP(NDIM,N,A,NPIV,IND)
C PARAMETERS

C NDIM - AN INTEGER INDICATING THE NUMBER OF ROWS IN THE
C ARRAY A AS DECLARED IN THE CALLING PROGRAM.

C N - AN INTEGER CONSTANT INDICATING THE SIZE OF THE SYSTEM
C TO BE SOLVED.

C A - A REAL 2-DIMENSIONAL ARRAY, OF SIZE NDIM*N, HOLDING
C THE MATRIX TO BE DECOMPOSED. ON RETURN, THE CONTENTS
C OF A ARE REPLACED BY THE LU FACTORIZATION.

C NPIV - AN INTEGER VECTOR, OF DIMENSION N, WHICH IS
C UNINITIALIZED AT THE TIME OF CALLING. THIS ARRAY
C WILL RECORD THE REARRANGING OF THE ROWS.

C IND - AN INTEGER INDICATING IF A IS SINGULAR OR NOT.
C IND = -1 A IS SINGULAR.
C = 0 A IS NONSINGULAR.

C THIS PARAMETER IS NOT INITIALIZED AT TIME OF CALLING.

C
C
C 0000

C SAMPLE CALLING PROGRAM
C DIMENSION A(5,4),NPIV(4)

C READ,N,((A(I,J)),J=1,N),I=1,N)

C CALL DCOMP(5,N,A,NPIV,IND)

C IF(IND.NE.0) PRINT,'MATRIX IS SINGULAR'

C DO 1 I=1,N

C IPIVOT=NPIV(I)

C PRINT,(A(IPIVOT,J)),J=1,N)

C 1 CONTINUE

C PRINT?

C 2 FORMAT('0'.6X,'PRESENT ROW ORIGINAL ROW')

C DO 3 I=1,N

C
C 0000

```

C          PRINT,NPIV(I),I          0000
C  2      CONTINUE                  0000
C          STOP                      0000
C          END                        0000
C  #DATA                             0000
C  M(= OF UNKNOWNNS - CONSTANT) 1. 2. 3. ETC. 0000
C                                     0000
C                                     0000
C  REFERENCES  ELEMENTARY NUMERICAL ANALYSIS 0000
C              BY CONTE AND DE BOOR.         0000
C                                     0000
5  DIMENSION A(NDIM,N),NPIV(N)      0000
6  IND = 0                             0000
C                                     0000
C  *****                          0000
C  *                                  0000
C  * CHECK FOR A SYSTEM OF ONLY ONE UNKNOWN 0000
C  *                                  0000
C  *****                          0000
C                                     0000
7  IF (N .EQ. 1) RETURN              0000
C                                     0000
C  *****                          0000
C  *                                  0000
C  * INITIALIZE PIVOT VECTOR           0000
C  *                                  0000
C  *****                          0000
C                                     0000
8  DO 10 I=1,N                       0000
9  NPIV(I)=I                          0000
10 10  CONTINUE                       0000
C                                     0000
C  *****                          0000
C  *                                  0000
C  * MAIN LOOP FOR GAUSS ELIMINATION  0000
C  *                                  0000
C  *****                          0000
C                                     0000
1  NM1=N-1                             0000
2  DO 22 I=1,NM1                       0000
C                                     0000
C  *****                          0000
C  *                                  0000
C  * SEARCH COLUMN FOR LARGEST PIVOT,I.E., 0000
C  * MAX A(J,I) , ( I )= J )= N.      0000
C  *                                  0000
C  *****                          0000
C                                     0000
3  COLMAX=0.                           0000
4  DO 14 J=I,N                         0000
5  HOLD=ABS(A(NPIV(J),I))              0000
6  IF (HOLD .LE. COLMAX) GO TO 14      0000
7  COLMAX=HOLD                         0000
8  NROW=J                               0000
9 14  CONTINUE                         0000
C                                     0000
C  *****                          0000
C  *                                  0000
C  * TEST FOR SINGULARITY. THE MATRIX IS ASSUMED TO BE SINGULAR 0000

```

```

C      * IF COLMAX (THE ABS. VALUE OF THE PIVOT) IS EQUIVALENT      0000000
C      * TO ZERO, I.E.,                                           0000000
C      *      1.0 + COLMAX = 1.0 .                                0000000
C      * IF THIS IS TRUE THEN THE ROUTINE PROCEEDS ON TO THE I+1)-TH 0000000
C      * STAGE OF THE ELIMINATION.                                0000000
C      *                                                            0000000
C      * *****                                                0000000
C      *                                                            0000000
20      IF (1.+COLMAX .NE. 1.) GO TO 16                            0000000
21      IND = -1                                                  0000000
22      GO TO 22                                                  0000000
C      * *****                                                0000000
C      *                                                            0000000
C      * IF AN INTERCHANGE IS NECESSARY, ALTER THE PIVOT VECTOR NPIV. 0000000
C      *                                                            0000000
C      * *****                                                0000000
C      *                                                            0000000
23      16      IPIVOT=NPIV(NROW)                                  0000000
24      IF (NROW .EQ. I) GO TO 17                                0000000
25      NPIV(NROW)=NPIV(I)                                       0000000
26      NPIV(I)=IPIVOT                                           0000000
C      * *****                                                0000000
C      *                                                            0000000
C      * THE MULTIPLIERS FOR THE COMPUTATION OF THE REMAINDING ROWS ARE 0000000
C      * DETERMINED AND ELIMINATION IS PERFORMED. THE VALUE OF EACH 0000000
C      * MULTIPLIER IS STORED IN THE POSITION OF THE ELIMINATED      0000000
C      * ELEMENT.                                                 0000000
C      *                                                            0000000
C      * *****                                                0000000
C      *                                                            0000000
27      17      IPI = I+1                                         0000000
28      DO 20 J=IPI,N                                             0000000
29      IPIVOT=NPIV(J)                                           0000000
30      AMULT=A(JPIVOT,I)/A(IPIVOT,I)                            0000000
31      A(JPIVOT,I)=AMULT                                        0000000
32      DO 18 K=IPI,N                                             0000000
33      A(JPIVOT,K)=A(JPIVOT,K)-AMULT*A(IPIVOT,K)              0000000
34      18      CONTINUE                                          0000000
35      20      CONTINUE                                          0000000
36      22      CONTINUE                                          0000000
37      IF (1.+ABS(A(NPIV(N),N)) .EQ. 1.) IND=-1                0000000
38      RETURN                                                    0000000
39      END                                                       0000000

```

ERROR INVALID TYPE OF ARGUMENT IN REFERENCE TO SUBPROGRAM DCOMP

PROGRAM WAS EXECUTING LINE 1 IN ROUTINE M/PRG WHEN TERMINATION OCCURRED

CORE USAGE OBJECT CODE= 1608 BYTES, ARRAY AREA= 0 BYTES, TOTAL AREA AVAILABLE= 170984 BYTES

DIAGNOSTICS NUMBER OF ERRORS= 1, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSIONS= 0

COMPILE TIME= 0.79 SEC, EXECUTION TIME= 0.00 SEC, WATFIV - JUL 173 V1L4 18.37.20 MONDAY 16 APR 79

```
//JANFIDE JOB MSGLEVEL=1
// EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=3330,VOL=SER=UEPVS2,DISP=OLD
//SYSIN DD *
//SYSIN DD * GENERATED STMT
IEF236I ALLOC. FOR JANFIDE
IEF237I 152 ALLOCATED TO DD1
* IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I SYS79106.T143628.RE101.JANFIDE.R0000001 KEPT
IEF285I VOL SER NOS= UEPVS2.
IEF298I JANFIDE SYSOUT=A.
```

DIRECTORY INFO FOR SPECIFIED PDS ON VOL UFPVS2
WATEIV.WATITR

MEMBERS	TTRC	VARIABLE USER DATA ---(USER DATA AND T ARE IN HEX)
ACCUM	00001602	01200000
ADFM	000800F02	00200000
AINTRD	00010000	
AINTRS	00011000	
ALICW	00020102	01200000
ANFRMD	00020400	
ANFRMS	00020600	
ANFVA	00020800	
ANVAMT	00020F00	
ARANK	00030100	
ASRANK	00030400	
BAIANC	004F0000	
BAIPAK	004F0F00	
PARID	00030600	
PARIS	00030800	
RFSJ	00030A00	
RFSK	00030C02	01200000
RFSY	00031100	
RTSEC	00940002	00200000
BLOCK	00031400	
ROCI	00040302	01200000
CAT001	00900402	00200000
CRARK2	00500900	
CBAL	00510400	
COMIN	00040602	00200000
CEIPI	009A0702	00200000
CEIPE	00990202	00200000
CEWOP	00040F02	01000000
CGMING	00041302	00200000
CHICHT	00050700	
CHICRP	00050000	
CHREDD	00060400	
CHRECN	00060600	
CIDE	00520900	
CITMR	00060802	01200000
CILSQ	00070702	01200000
CIDSE	00080800	
CMFERD	00080F02	01200000
CMXTNV	009B0102	00200000
CNTDHR	00090502	01000000
COMFES	00530700	
CCMIR2	00540600	
COACAT	000A0102	01200000
CONREP	000A0400	
CONTIN	00991002	00200000
CORCOE	00000400	
CPCE01	008F0102	00200000
CPM	00000600	
CQDF	00560000	
CVAR	000F0102	01200000
DAIAG	00570F00	

SYSTEMS SUPPORT UTILITIES---IFHLIST

DASHG	000E0302	01200000
DCIDE	00590500	
DCOMP	00590A00	
DCOMS	005A0B00	
DCODE	005B0E00	
DCRDM	005C0600	
DDCOMP	005E0600	
DDCOMS	005E0400	
DDPRDD	00600100	
DECOMP	000E0700	
DEPCUR	00600500	
DEPI IN	00600900	
DETMAD	000E0D00	
DETMAS	000E0F00	
DEVNWT	00600D00	
DI DE	00601000	
DI EP	000E1100	
DI EPS	000E1300	
DI N SQ	00610100	
DMEGCD	000E1500	
DMEGRD	000E0D00	
DMEGRS	00100C00	
DMGRAM	00610700	
DMODRE	00611000	
DMUI R	00620800	
DNEWST	00630100	
DNEWIN	00630900	
DOPRDD	00110B00	
DOPRDS	00110D00	
DOPRDN	00110F02	01200000
DOPENM	00130302	01200000
DPEFRM	00630F00	
DPITPN	00640500	
DPCFR	00150B02	01200000
DPRRM	00160D02	01200000
DPRDD	00640A00	
DPWI IN	00640F00	
DRAFT	00170F02	01200000
DRANK	00180A00	
DRAWS	00180D02	01200000
DRDM	00641200	
DSCIVE	00660500	
DSPIN	00660D00	
DSPINE	00670700	
DSRANK	001A0100	
ETCVAI	001A0302	01200000
ETCVFC	001B0302	01200000
ETCVRD	001E0100	
ETCVRS	00200400	
ETPCD	00210602	01200000
EIMFES	00670F00	
EITRAN	00680B00	
EPCUR	00690500	
EPI IN	00690A00	
FUI ER	00690F00	

SYSTEMS SUPPORT UTILITIES---LIST

FVCCN	00220100	
FVWLT	006A0600	
FACTAN	00230502	01200000
FAT	008D1202	00200000
FCC	00970002	00200000
FCN	009C0302	00200000
FDHINT	00240C02	01200000
FDHIST	00250902	03200000
FTVPAK	00260602	02200000
GAHQUD	00260F00	
GAHQUS	00270400	
GDATE	00270800	
GENCV	006A0B02	00200000
GENCVV	006B0502	00200000
GENRV	006B0F02	00200000
GENRVV	006C0502	00200000
GEPI SD	00270C02	01200000
GEPI SS	00270F00	
GJDAN	00930F02	00200000
GI GQUD	00271200	
GI GQUS	00280000	
GO	006D0600	
GRAPHIX	00290602	01200000
GRAPHO	002A0902	05200000
GRAPHI	002A1002	06200000
GRAPH3	002B0B02	05200000
GRAPH5	002B0F02	05200000
GSITND	002B1102	01200000
GSITNS	002C0100	
GSNFID	002C0400	
GSNFIS	002C0700	
HAMPCD	002C0A00	
HAMPCS	002C0C00	
HARMAD	002C0F00	
HARMAN	002C1000	
HESBRD	002C1200	
HESBRS	002D0500	
HESQR	002D0F02	01200000
HGROUP	002E0F02	01200000
HISTGM	002E0702	01200000
HQR	006E0200	
HQR2	00700B00	
IMPR	008D1402	00200000
IMPRV	00730B00	
INCRXM	002E0B00	
INDEX	002E0D02	01200000
INITP	002E1002	01200000
IRANK	002E1300	
ISRANK	002E1600	
ITAV1D	00300100	
ITAV1S	00300300	
ITAV2D	00300500	
ITAV2S	00300700	
JCFIGD	00300900	
JCFIGS	00300C00	

SYSTEMS SUPPORT UTILITIES---TEHLIST

KIDENT	008E0102	00200000
LARGEFP	00300F02	01200000
LDF	00740A00	
LEAST	00301102	01200000
LEF	00900302	00200000
LETTRS	00310600	
LEYR	009B0602	00200000
LISQ	00310A02	01200000
LINECD	00320402	01200000
LINEONS	00320702	01200000
LINISQ	00750900	
LISCFD	00320A00	
LISCFE	00320F00	
LITINC0	00321200	
LITINCS	00321500	
LITINRD	00330300	
LITINRS	00330600	
MAADDD	00330900	
MAADDS	00330B00	
MAFXPD	00330D00	
MAFXPS	00330F00	
MAMAXD	00331100	
MAMAXS	00331300	
MAMP1D	00331500	
MAMP1S	00331700	
MAMP2D	00331900	
MAMP2S	00331B00	
MASFTD	00331D00	
MASFTS	00331F00	
MASURD	00332100	
MASURS	00332300	
MATEQ	008E0502	00200000
MEANSD	00332500	
MERCFE	00332700	
MERCFE	00340402	01200000
MGRAM	00760500	
MINVCD	00340800	
MINVCS	00340B00	
MINVRD	00340F00	
MINVRS	00341100	
MI REGR	00341400	
MOPRE	00770500	
MOVE	00180082	01200000
MOVE2	00180082	01200000
MPICT	00360302	03200000
MUIR	00780500	
MUIRRD	00360800	
MUIERS	00360B00	
NBAM	00360F00	
NBAMD	00370400	
NDTSD	00370002	01200000
NDTSS	00370F02	01200000
NFWST	00790600	
NFWTC	00901102	00200000
NFWTN	007A0A00	

SYSTEMS SUPPORT UTILITIES---IFHIST

NFWTRD	00371100	
NFWTRS	00371300	
NFWTSD	00371500	
NFWTSS	00371700	
NFWTUD	00380100	
NFWTUS	00380300	
NI DECD	00380500	
NI DECS	00380700	
NI MAX	00380902	01200000
NI MAX1	00381202	01200000
NI PFAP	00390A00	
NMSERS	003B0400	
NRMING	003B0F00	
NUM	003C0F00	
OPTMUD	003C1002	00200000
OPTMUM	003E0000	
ORCFR	00420A00	
PCCOPYD	00420F00	
PCCPYS	00421000	
PHERM	00780900	
PITPN	007C0500	
PI OT	008E1102	00200000
PI OT1	00421200	
PI OT2	00430100	
PMU TD	00430400	
PMU TS	00430600	
PDIARG	00430802	01200000
PDIARN	00430E02	01200000
PDIAPP	00431102	01200000
PDIART	00440502	01200000
PDIYCD	00440800	
PDIYCP	00440A00	
PQFR	00440C02	01200000
PRRM	00450B02	01200000
PRIN	00990C02	00200000
PRDDU	008E0702	00200000
PWFICD	00460D02	01200000
PWFICS	00460F02	01200000
PWI IN	007C0E00	
QNMING	00461102	00200000
RATQUD	00470500	
RADQUS	00470900	
RANDI	00470C00	
RANDS	00470F00	
RRERG	00950A02	00200000
RKFESTP	007D0400	
RKCSI D	00471000	
RKCSI S	00471300	
RK5FS	00471600	
RMBRCD	00471900	
RMBRCS	00480200	
ROTATS	00480402	01200000
RUK4	00480802	01200000
SCFPLY	00480A00	
SCMAT	008E0902	00200000

SYSTEMS SUPPORT UTILITIES---IFHLIST

(quite) numbers 3 names etc. etc.

SCMPYD	00490A00	
SCMPYS	00490C00	
SCMIT	008F1902	00200000
SCUMT	008F1502	00200000
SEARCH	00490F00	
STMPRD	00491200	
STMPRS	00491400	
SOLVE	007F0402	00200000
SCMA	008F1702	00200000
SRRTA	004A0100	
SRRTB	004A0180	
SRRTID	004A0402	01200000
SRRTIS	004A0602	01200000
SPIINF	004A0800	
SPIN	007F1000	
SPINF	007F0F00	
STORFD	004A0800	
STORFS	004A0D00	
STINTX	00800D00	
SUPD	004A0F00	
SUPMCD	004A1100	
SUPMCS	004A1300	
SUPMXD	004A1500	
SUPMXS	004A1700	
SUPSTR	004A1902	01200000
SVD	00810A00	
SYMRV	00840502	00200000
SYMRVV	00840F02	00200000
TSTF	008A0F02	00200000
THREFD	004A1C00	
TQ11	00850800	
TQ12	00860800	
TRACFD	004B0C00	
TRACES	004B0F00	
TREDF1	00870C00	
TREDF2	00880A00	
TREND	004B1002	03200000
TRNSPD	004C0700	
TRNSPS	004C0900	
TREFTP	00890800	
UNPACK	004C0800	
VALMX	008F0F02	00200000
VCMPYD	004C0F00	
VCMPYS	004C1100	
VEPRDD	004C1300	
VEPRDS	004C1500	
VESFTD	004C1700	
VESEFS	004C1900	
WJNF	004C1800	
WJISD	004D0102	01200000
ZTRMCD	004E0600	
ZYMBEI	004F0802	05200000

IF THE 00020 DIRECTORY BLOCKS ALLOCATED TO THIS PDS, 00002 ARE(IS) COMPLETELY UNUSED

MEMO