

mgr CELINA SŁODOWY
Przemysłowy Instytut
Automatyki i Pomiarów MERA-PIAP
Warszawa

PAKIET PODPROGRAMÓW MATEMATYCZNYCH – PPM

1. Krótka charakterystyka i zawartość pakietu PPM

W pracach wymagających stosowania aparatu matematycznego analizy, algebry, rachunku całkowego, statystyki matematycznej, rachunku macierzowego itp. z koniecznością prowadzenia obliczeń numerycznych na maszynie cyfrowej, wygodnym narzędziem jest Pakiet Podprogramów Matematycznych PPM zainstalowany na maszynie cyfrowej R-32 w Laboratorium Obliczeń i Modelowania w MERA-PIAP. Pakiet ten jest zbiorem bibliotecznym, dołączonym do grupy bibliotek systemowych i stanowi rozszerzenie podstawowej biblioteki funkcji języka Fortran. Zawiera podprogramy napisane w języku Fortran, przystosowane do włączania do programów pisanych w tym języku. Podprogramy PPM charakteryzują się następującymi cechami:

- 1) zawierają jedynie instrukcje opisujące algorytm obliczeń danego zagadnienia,
- 2) nie mają narzuconych stałych wymiarów tablic dla danych wejścia/wyjścia,
- 3) używane w nich macierze symetryczne i diagonalne mogą być zapamiętane w sposób zapewniający oszczędne gospodarowanie pamięcią operacyjną.

Podprogramy wchodzące w skład pakietu PPM nie różnią się od podprogramów typu SUBROUTINE definiowanych przez użytkowników; wszystkie z wyjątkiem RECP*) są wywoływane za pomocą standardowego zdania CALL. Wywołanie podprogramu powinno być poprzedzone zadeklarowaniem wymiarów tablic i nadaniem wartości zmiennym wejściowym.

Dokumentacja pakietu PPM zawiera skrótowy opis każdego podprogramu i podaje:

- cel użycia podprogramu,
- sposób wywołania,
- opis parametrów,
- ograniczenia,
- krótkie informacje o wykorzystywanej metodzie obliczeń.

Do dyspozycji użytkowników jest wydruk postaci źródłowej podprogramów zawartych w pakiecie PPM. Pakiet PPM zawiera 221 podprogramów mających zastosowanie przy rozwiązywaniu następujących zagadnień:

- 1) z dziedziny statystyki matematycznej:
 - analiza wariancji,
 - analiza korelacji,
 - wieloraka regresja liniowa,
 - regresja krokowa,
 - regresja wielomianowa,
 - korelacje kanoniczne,
 - analiza czynnikowa,
 - analiza dyskryminacyjna,

*) RECP – działa jak Fortranowska funkcja i może być użyty jako argument przez inny podprogram

- analiza szeregów chronologicznych,
- analiza danych,
- nieparametryczne testy istotności,
- generacje liczb losowych o rozkładzie normalnym i jednostajnym,
- funkcje rozkładów prawdopodobieństwa,
- z dziedziny algebry macierzy:
 - odwracanie macierzy,
 - wartości własne i wektory własne,
 - transponowanie macierzy,
 - rachunek macierzowy,
 - elementarne przekształcenia macierzy,
- z dziedziny wielomianów i układów równań liniowych:
 - arytmetyka wielomianów,
 - określanie wartości wielomianów, całkowanie, różniczkowanie,
 - znajdowanie pierwiastków rzeczywistych i zespolonych dla wielomianów,
 - wielomiany: Legendre'a, Laguerre'a, Hermite'a, Czebyszewa,
 - układy algebraiczne równań liniowych,
- z dziedziny analizy matematycznej:
 - całkowanie i różniczkowanie funkcji w postaci wyrażenia lub tablicy,
 - rozwiązywanie układów równań różniczkowych zwyczajnych pierwszego rzędu,
 - analiza Fouriera dla funkcji Bessele'a lub modyfikowanych funkcji Bessele'a,
 - określenie wartości funkcji Gamma,
 - eliptyczne funkcje Jacobi'ego,
 - całki eliptyczne, wykładnicze, całki Fresnele'a,
 - aproxymacje, interpolacje i tworzenie tablic,
 - znajdowanie pierwiastków rzeczywistych dla zadanych funkcji.

2. Uwagi praktyczne dla użytkowników

Uruchomienie jakiegokolwiek programu na maszynie cyfrowej R-32 z systemem operacyjnym OS/JS, a więc i programu napisanego w języku Fortran z wykorzystaniem pakietu PPM, wymaga dostarczenia systemowi operacyjnemu opisu pracy jaka ma zostać wykonana. Opis taki, przygotowany w postaci zestawu zdań Języka Opisu Prac (JCL) wraz z programem źródłowym i danymi, stanowi dla systemu jednostkę przetwarzania, tzw. JOB.

Praca polegająca na realizacji programu w języku Fortran składa się z trzech faz odpowiadających krokom w Języku Opisu Prac JCL:

- kompilacja programu źródłowego – (krok FORT)
- łączenie – (krok LKED)
- załadowanie do PAO i wykonanie – (krok GO)

Realizację programu napisanego w języku Fortran można zaplanować według jednego z podanych poniżej wariantów.

Wariant W1

- A. Wykonać samą kompilację programu.
- B. Jeśli kompilacja wykonana poprawnie, a program będzie wielokrotnie wykorzystywany, wykonać kroki kompilacji i łączenia, a moduł ładowalny odesłać do prywatnej biblioteki użytkownika.
- C. Jeżeli użytkownik nie posiada prywatnej biblioteki modułów ładowalnych, można ją w tym momencie założyć i wykonać wszystkie czynności podane w pkt B.

D. Wywołać program (moduł ładowny) z prywatnej biblioteki użytkownika, dostarczyć mu dane i wykonać go.

Przypadek D można powtarzać dowolną ilość razy dostarczając programowi różnych zestawów danych.

Wariant W2

Wykonać pracę kompleksową polegającą na realizacji wszystkich trzech faz w jednym JOB-ie, co oznacza kompilowanie, łączenie i wykonanie programu w trakcie jednego przebiegu. Nie jest to ekonomiczny sposób wykorzystywania maszyny cyfrowej, zwłaszcza dla programów znajdujących wielokrotne zastosowanie.

Podane dalej przykłady ilustrują, w jaki sposób należy przygotować zestawy zdań JCL w każdym z przypadków.

W podanych zestawach zdań JCL należy zastąpić ciągi znaków pisanych małymi literami przez odpowiedni parametr lub nazwę nadaną przez użytkownika.

Pr z y p a d e k W1A

Praca polega na wywołaniu kompilatora języka Fortran, dla którego program źródłowy użytkownika stanowi dane. W tym celu należy posłużyć się jednokrokową procedurą skatalogowaną o nazwie FORTGC stosując następujący zestaw zdań JCL:

```
//aaa JOB MSGLEVEL=1
//bbb EXEC FORTGC
//SYSIN DD *
```

```
    } karty z programem źródłowym użytkownika w języku Fortran
```

```
/*
```

```
//
```

gdzie:

aaa — identyfikator pracy, czyli nazwa JOB-u,

bbb — nazwa kroku (może nie występować, wtedy spacje)

Pr z y p a d e k W1B

Praca polega na kompilacji i łączeniu programu, przy czym zostają dołączone odpowiednie moduły z biblioteki zawierającej podprogramy PPM. Powstały w wyniku tych operacji tzw. moduł ładowny użytkownika zostaje odesłany do jego prywatnej biblioteki.

W podanym zestawie zdań JCL wywołana zostaje procedura skatalogowana FORTGCL, w której do kroku łączenia (krok LKED) zostają dołączone biblioteki:

systemowa — SYS1.FORTLIB

podprogramów PPM — MODMAT.ELWRO

prywatna użytkownika — ppp

Zestaw zdań JCL dla tego przypadku wygląda następująco:

```
//aaa JOB MSGLEVEL=1
//bbb EXEC FORTGCL
//FORT.SYSIN DD *
```

```
    } karty z programem źródłowym użytkownika w języku Fortran
```

```
/*
```

```
//LKED.SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR
// DD DSN=MODMAT.ELWRO,DISP=SHR
```

```
//LKED.SYSLMOD DD DSN=ppp(mmm),
// UNIT=2314,VOL=SER=ddd,DISP=OLD
//
```

gdzie:

```
aaa — }
bbb — } jak w przypadku W1A
```

ppp — nazwa biblioteki użytkownika

mmm — nazwa identyfikująca moduł ładowalny w bibliotece użytkownika

ddd — etykieta dysku na jakim znajduje się biblioteka użytkownika

P r z y p a d e k W1C

Praca polega na kompilacji i łączeniu programu jak w przypadku W1B, a ponadto na założeniu biblioteki prywatnej, do której zostaje odesłany moduł ładowalny użytkownika.

Należy uprzednio uzgodnić z programistą systemowym na jakim dysku biblioteka może być zaalokowana i jaki obszar zostanie jej przydzielony. Analogicznie jak w przypadku W1B jest wywoływana procedura skatalogowana FORTGCL, a opis pracy wygląda następująco:

```
//aaa JOB MSGLEVEL=1
//bbb EXEC FORTGCL
//FORT.SYSIN DD *
```

```
}
} karty z programem źródłowym użytkownika w języku Fortran
```

/*

```
//LKED.SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR
// DD DSN=MODMAT.ELWRO,DISP=SHR
//LKED.SYSLMOD DD DSN=ppp(mmm),UNIT=2314,VOL=SER=ddd,
// DISP=(NEW,KEEP),SPACE=(kkk,(I1,I2,I3),,CONTIG)
//
```

gdzie:

```
aaa — }
bbb — }
ppp — } jak w przypadku W1B
mmm — }
ddd — }
```

kkk — jednostki w jakich będzie dokonane rezerwowanie obszaru. Można podstawić: TRK — alokacja w ścieżkach, CYL — alokacja w cylindrach.

I1 — alokacja pierwotna (obszar wstępnie przydzielony),

I2 — alokacja wtórna (rozszerzenie obszaru pierwotnego do 15-tu razy),

I3 — ilość bloków przeznaczonych na skorowidz biblioteki

P r z y p a d e k W1D

Moduł ładowalny programu użytkownika zostaje wywołany z jego prywatnej biblioteki, po wprowadzeniu go do PAO i dostarczeniu danych zostaje wykonany.

Dla tego przypadku stosuje się następujący zestaw zdań JCL:

```
//aaa JOB MSGLEVEL=1
//JOBLIB DD DSN=ppp,UNIT=2314,VOL=SER=ddd,DISP=OLD
//bbb EXEC PGM=mmm
//FT03F001 DD SYSOUT=A
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
```

```
... } dane do programu
```

```
/*
```

```
//
```

```
gdzie:
```

```
aaa - }  
bbb - } jak w przypadkach: W1B i W1C  
ddd - }  
mmm - }  
ppp - }
```

Przypadek W2

Zostaje wywołana procedura skatalogowana FORTGCLG umożliwiająca wykonanie w jednym JOB-ie wszystkich trzech faz koniecznych do osiągnięcia wyników działania programu.

Zestaw zdań JCL dla tego przypadku:

```
//aaa JOB MSGLEVEL=1
```

```
//bbb EXEC FORTGCLG
```

```
//FORT.SYSIN DD *
```

```
... } karty z programem źródłowym użytkownika w języku Fortran  
... }
```

```
/*
```

```
//LKED.SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR
```

```
// DD DSN=MODMAT.ELWRO,DISP=SHR
```

```
//GO.SYSIN DD *
```

```
... } karty z danymi dla programu  
... }
```

```
/*
```

```
//GO.FT03F001 DD SYSOUT=A
```

```
//
```

```
gdzie:
```

```
aaa - }  
bbb - } jak w poprzednich przypadkach
```