Znanstveno računanje 1

1. dio vježbi
LAPACK

Nela Bosner

LAPACK

- Ime LAPACK je kratica za *Linear Algebra PACKage*.
- LAPACK je biblioteka potprograma pisanih u Fortranu 77 za rješavanje najčešćih problema u numeričkoj linearnoj algebri.
- Dizajniran je da efikasno radi na širokom rasponu modernih i brzih računala.
- LAPACK može riješiti
  - sustave linearnih jednadžbi
  - linearni problem najmanjih kvadrata
  - problem svojstvenih vrijednosti
  - problem singularnih vrijednosti
- uključuje joši i računanje raznih pomoćnih problema i veličina, kao npr.: matrične faktorizacije i procjene broja uvjetovanosti matrica.
- Sva dokumentacija i software dostupan je na

http://www.netlib.org/lapack/

- Biblioteka se sastoji od
  - **upravljačkih (driver) potprograma** za rješavanje standardnih problema
  - **računskih (computational) potprograma** za rješavanje određenih računskih zadataka
  - **pomoćnih (auxiliary) potprograma** za izvođenje određenih podzadataka ili računanje osnovnih operacija
- Svaki upravljački potprogram obično poziva niz računskih potprograma.
- Računski potprogrami sveukupno mogu rješavati širi raspon problema nego upravljački.
- Pomoćni potprogrami obično se koriste za numeričku analizu.
- Gotovo sve rutine postoje u **realnoj** i **kompleksnoj** varijanti, i to u **single** i **double** preciznosti.
- LAPACK omogućuje rad sa gusto popunjenim i vrpčastim matricama.

- LAPACK potprogrami su pisani tako da se što je moguće više osnovnih operacija izvodi pozivanjem BLAS (*Basic Linear Algebra Subprograms*) potprograma.
- Vrlo efikasni BLAS potprogrami su implementirani u ovisnosti o platformi na kojoj se izvode, i dostupni su za većinu modernih računala.
- BLAS nije dio LAPACK-a jer su LAPACK potprogrami neovisni o platformi, ali mu omogućuju vrhunsku učinkovitost.
- Dokumentacija i software dostupan je na

  `http://www.netlib.org/blas`

- Postoje tri stupnja BLAS potprograma:
  1. BLAS 1 potprogrami za vektorske operacije, kao npr.:
  $$y \leftarrow \alpha x + y$$
  2. BLAS 2 potprogrami za matrično–vektorske operacije, kao npr.:
  $$y \leftarrow \alpha A x + \beta y$$
  3. BLAS 3 potprogrami za matrične operacije, kao npr.:
  $$C \leftarrow \alpha A B + \beta C$$
- BLAS 3 potprogrami su najefikasniji jer se vrši $\mathcal{O}(n^3)$ operacija s pomičnom točkom nad $\mathcal{O}(n^2)$ podataka, dok BLAS 2 vrši samo $\mathcal{O}(n^2)$ operacija nad $\mathcal{O}(n^2)$ podataka.

Pravilo imenovanja LAPACK potprograma:

- Svi upravljački i računski potprogrami imaju imena oblika

  XYYZZZ

- X predstavlja tip podataka

S	REAL
D	DOUBLE PRECISION
C	COMPLEX
Z	COMPLEX*16 ili DOUBLE COMPLEX

- YY predstavljaju tip matrice, kao npr.:

BD	bidijagonalna
DI	dijagonalna
GB	opća vrpčasta
GE	opća (nesimetrična, i ponekad pravokutna)
HE	hermitska
OR	ortogonalna
PO	simetrična ili hermitska pozitivno definitna
SY	simetrična
TR	trokutasta
UN	unitarna

  kod većine pomoćnih potprograma drugo i treće slovo je LA.
- ZZZ predstavljaju operaciju ili račun koja se izvršava

Primjer

Upravljački potprogram za rješavanje općenitog sustava linearnih jednadžbi zove se SGESV:

$$SGESV$$

pri čemu je

S	*ulazni i izlazni parametri su realne varijable single preciznosti*
GE	*matrica sustava je općenita gusto popunjena matrica*
SV	*rješava se sustav linearnih jednadžbi*

Napomena: Treće slovo kod ZZZ može biti prazno.

CLAPACK

- C verzija LAPACK-a.
- CLAPACK biblioteka napravljena je pomoću alata za prebacivanje Fortran koda u C, nazvan *f2c*.
- Moraju se poštivati Fortranovska pravila za pozivanje potprograma i Fortranovske strukture podataka.

1. Imena C potprograma moraju se razlikovati od istovjetnih Fortran potprograma.
   - Fortranovskom imenu potprograma dodaje se na kraju donja crtica (_)

Fortran	**C**
`CALL SGESV(···)`	`sgesv_(···)`

2. Argumenti potprograma moraju se prenjeti po adresi, tj. kao pokazivači.

Fortran	**C**
`CALL SGESV( 5, 1, A, 5,` `IPIV, B, 5, INFO )`	`N=LDA=LDB=5; NHRS=1;` `sgesv_( &N, &NRHS, A, &LDA, IPIV, B, &LDB, &INFO )`

3 Dvodimenzionalna polja se različito definiraju u C-u i Fortranu.

- dvodimenzionalno polje u **Fortranu**

```
DOUBLE PRECISION A(LDA, N)
```

je kontinuirani komad memorije sastavljen od LDA$\times$N riječi u double preciznosti, spremljenih po stupcima koji se nalaze jedan iza drugog

- dvodimenzionalno polje u **C**-u

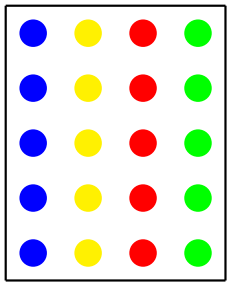
```
double A[LDA][N];
```

je niz od LDA pokazivača na redove duljine N, u memoriju se spremaju redovi koji ne moraju biti jedan za drugim

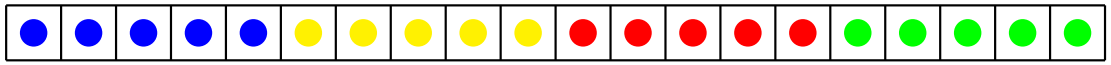
- za pozivanje CLAPACK potprograma treba koristiti jednodimenzionalno polje veličine LDA$\times$N riječi u double preciznosti

```
double *A;
A = malloc( LDA*N*sizeof(double) );
```

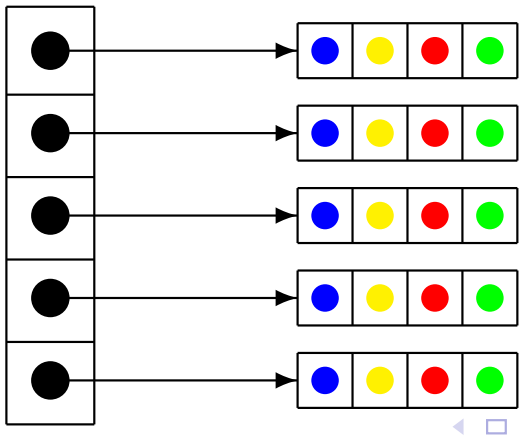
Matrica:

$A =$ 

Fortran:

A: 

C:

A: 

Primjer (Fortranov pristup poljima u C-u)

Sljedeći dio koda prikazuje kako inicijalizirati $M \times N$ polje `A` *tako da je element na poziciji* (i, j) *jednak* $i + j$.

```
double *A;
A = malloc( M*N*sizeof(double) );
for(j=0; j < N; j++)
{
  for (i=0; i < M; i++) A[j*M+i] = i+j;
}
```

Zaglavlja (headers) potrebna za korištenje CLAPACK potprograma:

- `f2c.h` — definicije raznih tipova podataka, kao npr.:

```
typedef long int integer;
typedef float real;
typedef double doublereal;
typedef struct { real r, i; } complex;
typedef struct { doublereal r, i; } doublecomplex;
typedef long int logical;
```

- `fblaswr.h` — deklaracija BLAS potprograma
- `clapack.h` — deklaracije CLAPACK potprograma

Primjer

Rješimo sustav $Ax = b$, pri čemu su

$$A = \begin{bmatrix} 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \\ 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 10 \\ 7 \\ 9 \end{bmatrix},$$

pomoću CLAPACK potprograma `dgesv_()`*. Izračunato rješenje ćemo usporediti s egzaktnim rješenje koje iznosi* $x = [\ 1 \quad 1 \quad 1 \quad 1\ ]^T$.

- *Potprogram* `dgesv_()` *računa rješenje realnog sustava linearnih jednadžbi $Ax = b$, gdje je A $n \times n$ matrica, a x i b su $n \times$ nrhs matrice.*
- *Za računanje je korištena LU faktorizacija s parcijalnim pivotiranjem*

Primjer (nastavak)

U zaglavlju `clapack.h` *deklaracija za taj potprogram je*

```
int dgesv_(integer *n, integer *nrhs,
doublereal *a, integer *lda, integer *ipiv,
doublereal *b, integer *ldb, integer *info);
```

pri čemu su:

`n`	*(ulaz) red matrice A*
`nrhs`	*(ulaz) broj stupaca matrice b*
`a`	*(ulaz) $n \times n$ matrica sustava A*
	(izlaz) faktori L i U, bez dijagonale od L
`lda`	*(ulaz) vodeća dimenzija polja* `a` *(lda $\geq$ n)*
`ipiv`	*(izlaz) n polje za spremanje indeksa koji definiraju matricu permutacija P:*
	redak i matrice A zamijenjen je retkom `ipiv`[*i*]

Primjer (nastavak)

`b`	*(ulaz) $n \times$ nrhs matrica desne strane b*
	(izlaz) $n \times$ nrhs matrica rješenja x
`ldb`	*(ulaz) vodeća dimenzija polja b (ldb $\geq n$)*
`info`	*(izlaz) informacija o izvršavanju potprograma (0=OK)*

- *Program `primjer1.c` koji rješava zadani problem i potrebna zaglavlja nalaze se na*

  `http://www.math.hr/~nela/zr1.html`

- *Program se kompajlira sa*

```
$ gcc primjer1.c -lblas -llapack
```

Program (primjer1.c)

```
#include <stdio.h>
#include <stdlib.h>
#include "f2c.h"
#include "fblaswr.h"
#include "clapack.h"
main(integer argc, char *argv[])
{
    doublereal a[]={1.0,2.0,5.0,1.0, 1.0,1.0,1.0,4.0, 4.0,1.0,1.0,1.0,
        1.0,6.0,0.0,3.0};
    doublereal b[]={7.0,10.0,7.0,9.0};
    doublereal x[]={1.0,1.0,1.0,1.0};
    doublereal alpha, rg;
    integer n, nhrs, *ipiv, info, incx;
    n=4;
    ipiv=malloc(n*sizeof(integer));
    nhrs=1;
    incx=1;
    alpha=-1.0;
    dgesv_( &n, &nhrs, a, &n, ipiv, b, &n, &info );
    printf("dgesv() je izracunao :[ %19.16f, %19.16f, %19.16f, %19.16f ]^T\n",
        b[0],b[1],b[2],b[3]);
    printf("Egzaktno rjesenje je :  [ 1, 1, 1, 1 ]^T\n");
    daxpy_( &n, &alpha, x, &incx, b, &incx );
    printf("Razlika izracunatog i egzaktnog rjesenja :[ %.2e, %.2e, %.2e, %.2e ]^T
        \n", b[0],b[1],b[2],b[3]);
    rg=dnrm2_( &n, b, &incx )/dnrm2_( &n, x, &incx );
    printf("Relativna greska rjesenja iznosi :  %.2e\n", rg);
}
```

- Matrice u primjenama su često velikih dimenzija, zato se ne preporuča unos elemenata matrica sa standardnog ulaza.
- Matrice se mogu unijeti
  - iz datoteke
  - generiranjem pomoću LAPACK potprograma
- CLAPACK potprogrami za generiranje matrica su:
  - `int dlaset_(char *uplo, integer *m, integer *n, doublereal *alpha, doublereal *beta, doublereal *a, integer *lda);`
  - `int dlarnv_(integer *idist, integer *iseed, integer *n, doublereal *x);`

- `int dlaset_()` inicijalizira $m \times n$ matricu A tako da elementi na dijagonali budu jednaki β a van dijagonale α.

`uplo`	(ulaz) određuje dio matrice A koji će biti inicijaliziran: = 'U': gornji trokut = 'L': donji trokut inače: cijela matrica
`m`	(ulaz) broj redaka matrice A
`n`	(ulaz) broj stupaca matrice A
`alpha`	(ulaz) konstanta α, vrijednost vandijagonalnih elemenata
`beta`	(ulaz) konstanta β, vrijednost dijagonalnih elemenata
`a`	(ulaz/izlaz) matrica A
`lda`	(ulaz) vodeća dimenzija polja `a` ($lda \geq m$)

- `int dlarnv_()` vraća vektor od n slučajnih brojeva iz unformne ili normalne distribucije.

  `idist` (ulaz) određuje distribuciju slučajnih brojeva:
  = 1: uniformna $\langle 0, 1\rangle$
  = 2: uniformna $\langle -1, 1\rangle$
  = 3: normalna $N(0, 1)$

  `iseed` (ulaz/izlaz) polje od 4 elemenata, sjeme generatora slučajnih brojeva;
  elementi polja moraju biti između 0 i 4095,
  a `iseed`[3] mora biti neparan;
  kod izlaza sjeme se ažurira

  `n` (ulaz) broj slučajnih brojeva koje treba generirati

  `x` (izlaz) polje generiranih slučajnih brojeva

Zadaci

Zadatak

Napišite program koji učitava broj n, te zatim generira $n \times n$ matricu A i n-dimenzionalan vektor x sa slučajnim brojevima.

1. *Sami napišite potprogram* `mojmv()` *za množenje matrice s vektorom, i primijenite ga na $A \cdot x$.*
2. *Primijenite BLAS potprogram* `dgemv_()` *za $A \cdot x$.*

U oba slučaja mjerite vrijeme izvršavanja potprograma za množenje matrice i vektora pomoću funkcije `clock()`. *Deklaracija* `clock_t clock(void);` *nalazi se u zaglavlju* `time.h`. *Vrijednost koju vraća ta funkcija dijeli se sa konstantom* `CLOCKS_PER_SEC` *čime se dobiva broj sekundi. Provjerite vaš program na širokom rasponu dimenzija. Što primijećujete?*

Zadatak

Napišite program koji učitava broj n, te zatim generira dvije $n \times n$ matrice A i B sa slučajnim brojevima.

1. *Sami napišite potprogram* `mojmm()` *za množenje dviju matrica, i primijenite ga na $A \cdot B$.*
2. *Primijenite BLAS potprogram* `dgemm_()` *za $A \cdot B$.*

U oba slučaja mjerite vrijeme izvršavanja potprograma za množenje matrica pomoću funkcije `clock()`*. Provjerite vaš program na širokom rasponu dimenzija. Što primijećujete?*