



07-Posebne biblioteke

POTPORNI SOFTVER ZA PERFORMANTNE NAUČNO RAČUNARSTVO

Zašto?

- ▶ Zašto koristimo biblioteke inače? Neko ko se *samo* bavi metodama za, npr. linearnu algebru može da ih implementira mnogo bolje od nas kojima je ta linearna algebra (potencijalno) samo jedna od deset stvari koje nam trebaju.
- ▶ Ovo je, u stvari, naročito dobar primer za upotrebu biblioteka zato što vrlo često mi imamo jasnu podelu između našeg algoritma i raznih matematičkih operacija i algoritama koji nam trebaju da ga implementiramo. Integracija, uz tako jasnu podelu, je mnogo lakša.
- ▶ Često ovo ide do tačke da se koristi poseban jezika (Python, često) ali mi nećemo nužno ići tako daleko.

Izbor iz oblasti primene

- ▶ Linearna algebra
- ▶ Parcijalne diferencijalne jednačine
- ▶ Grafovi
- ▶ Paralelizovani I/O
- ▶ Dekompozicija meš struktura
- ▶ Vizuelizacija

Struktura predavanja

- ▶ Prelazimo prvo najizraženiji deo HPC-a: linearnu algebru.
- ▶ U okviru HPC-a BLAS i Lapack se analiziraju u malo više detalja kao demonstracija kako izgledaju biblioteke tog tipa.
- ▶ Za ostale oblasti primene, samo se informativno prelaze koje biblioteke postoje i, konsekventno, odakle treba početi.

Linearna algebra

BLAS

- ▶ Basic Linear Algebra Subprograms
- ▶ BLAS je jedna od najdrevnijih biblioteka koje se još koriste.
- ▶ Nastala je kao deo rada u JPL ogranku NASA.
- ▶ BLAS je serijski i sastoji se od implementacija bazičnih formi vektorsko/matričnih operacija na način koji je jako optimizovan za različite platforme.

Nivoi

- ▶ Sve BLAS operacije se mogu podeliti na 3 nivoa po svojoj prirodi:
 1. Vektor-vektor
 2. Matrica-vektor
 3. Matrica-matrica
- ▶ Ranije verzije su podržavale samo niže nivoe zbog ograničenja računara iz tog doba.

Nivo	Tipizujuća jednačina
1	$y = \alpha x + y$
2	$y = \alpha Ax + \beta y$
3	$C = \alpha AB + \beta C$

Implementacija

- ▶ BLAS je napisan u programskom jeziku Fortran 77.
- ▶ Postoji varijanta prilagođena za upotrebu u programskom jeziku C koja se zove CBLAS
- ▶ Nju koristimo (danas, nju skoro svi koriste, kad koriste BLAS) te ako negde piše 'BLAS' gotovo se sigurno misli na CBLAS biblioteku.

Struktura naziva rutina

- ▶ Funkcije u BLAS-u imaju imena standardizovanog oblika koje, preko afiksa, određuju:
 - ▶ Preciznost
 - ▶ Tip matrice koji se koristi, ako ga ima.
 - ▶ Tip operacije.
- ▶ U skladu sa c konvencijom, postoji i prefiks biblioteke, cblas_

Preciznost u BLAS-u

Prefiks	Opis
s	Float
d	Double
c	Complex (dva float-a, 8 bajtova)
z	D. Complex (dva double-a, 16 bajtova)

Tip matrice u BLAS-u

Prefiks	Opis
ge	Opšta matrica: nema ograničenja.
sy	Simetrična matrica. Matrica koja je jednaka svojoj transpoziciji.
he	Hermitijanska matrica. Matrica u kojoj element-parovi tokom transponovanja su jedni drugima kompleksni konjugati.
tr	Trouglasta matrica gde su sve vrednosti ili iznad ili ispod glavne dijagonale nula.

Tip čuvanja matrice u BLAS-u

- ▶ Matrice se podrazumevano čuvaju u 'dense' formatu, odnosno, jednostavno se pamte sve vrednosti matrice. Redom.
- ▶ Ali, može se drugo slovo tipa matrice zameniti sa ili b ili p da odredi da je matrica drugog tipa.

Tip čuvanja matrice u BLAS-u

Prefiks	Opis
b	Trakasta (banded) matrica. Matrica u kojoj je većina vrednosti 0, ali one koje nisu su u dijagonalama (trakama).
p	Nabijena (packed) matrica. Ako je matrica takva da se od jednog trougla može rekonstruisati cela matrica (simetrična, Hermitijanska, ili trouglasta) može se čuvati samo taj jedan trougao pročitati po kolonama u jednom velikom nizu.

BLAS, nivo 1, operacije bez skalarnog proizvoda

Ime	Opis	Nivoi preciznosti
swap	Menja mesta vektorima	s,d,c,z
scal	Skalira vektor za konstantu	s,d,c,z,cs,zd
copy	Kopira vektor	s,d,c,z
axpy	Izvršava $\mathbf{y} = \alpha \mathbf{x} + \mathbf{y}$	s,d,c,z

BLAS, nivo 1, operacije sa skalarnim proizvodom

Ime	Opis	Nivoi preciznosti
dot	Skalarni proizvod	s,d,ds
dotc	Skalarni proizvod konjugovane vrednosti kompleksne vrednosti i neke druge kompleksne vrednosti	c,z
dotu	Kompleksni skalarni proizvod	c,z
sdsdot	Skalarni proizvod plus skalar	sds

BLAS, nivo 1, operacije norme

Ime	Opis	Nivoi preciznosti
nrm2	Proračun 2-norme $\ x\ _2 = \sqrt{\sum x_i ^2}$	s,d,sc,dz
asum	Proračun 1-norme $\ x\ _1 = \sqrt{\sum x_i }$	s,d,sc,dz
i_amax	Proračun ∞ -norme $\ x\ _\infty = \max(x_i)$	s,d,c,z

BLAS, nivo 1, operacije rotacije

Ime	Opis	Nivoi preciznosti
rotg	<p>Uz date skalare a i b, sračunaće c i s takve da:</p> $\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sqrt{ a ^2 + b ^2} \\ 0 \end{pmatrix}$ <p>Ovo se zovu paramteri Givensove rotacije</p>	s,d
rot	<p>Primenjuje Givens rotaciju, drugim rečima, ako joj se daju dva vektora kao ulaz, svaki element vektora se transformiše na sledeći način:</p> $x_i = cx_i + sy_i$ $y_i = -sx_i + cy_i$	s,d

BLAS, nivo 1, operacije rotacije

Ime	Opis	Nivoi preciznosti
rotmg	<p>Proračuna modifikovanu Givensovu rotacionu matricu. Ako su dati skalirajući faktori d1 i d2 i koordinate (x1, y1) ulaznog vektora sračunati 2x2 matricu H takvu da:</p> $\begin{pmatrix} x_1 \\ 0 \end{pmatrix} = H \begin{pmatrix} x_1 \sqrt{d_1} \\ y_1 \sqrt{d_2} \end{pmatrix}$	s,d
rotm	<p>Za vektore x i y proračuna:</p> $\begin{pmatrix} x_i \\ y_i \end{pmatrix} = H \begin{pmatrix} x_i \\ y_i \end{pmatrix}$	s,d

BLAS, nivo 2 i 3, operacije

Ime	Opis
mv	Proizvod matrice i vektora.
sv	Rešava matricu (trouglastu), tj. sistem linearnih jednačina koji ona predstavlja.
mm	Proizvod dve matrice uz učešće skalara.
rk	Računa opštu jednačinu: $C = \alpha AA^T + \beta C$
r2k	Računa opštu jednačinu: $C = \alpha AB^T + \bar{\alpha} BA^T + \beta C$

Primer, prost

- ▶ Setite se, kako davno, kada smo radili benchmarking HPC sistema.
- ▶ Pominjao sam da postoji jako često korišćen DGEMM korak benchmark-a.
- ▶ D-double preciznost
- ▶ GE-opšta matrica
- ▶ MM-množenje
- ▶ Funkcija bi bila `cblas_dgemm`

cblas_dgemm

```
1 void clblas_dgemm(  
2     const enum CBLAS_ORDER order,  
3     //matrice u redovi-prvo ili kolone prvo formi  
4     //CblasRowMajor ili CblasColMajor  
5     const enum CBLAS_TRANSPOSE transB,  
6     //Da li matricu A treba transponovati  
7     //CblasNoTrans, CblasTrans, CblasConjTrans  
8     const int M,  
9     //Broj redova u A i C  
10    const int N,  
11    //Broj kolona u B i C  
12    const int K,  
13    //Broj kolona u matrici A i broj redova u matrici B  
14    const double alpha,  
15    //Skalirajuci faktor za A*B
```

cblas_dgemm

```
16     const double *A,  
17     //Pokazivac na podatke za A  
18     const int lda,  
19     //Velicina prve dimenzije matrice A  
20     const double *B,  
21     //Pokazivac na podatke za B  
22     const int ldb,  
23     //Velicina prve dimenzije matrice B  
24     const double beta,  
25     //skalirajuci faktor za C  
26     double *C,  
27     //Pokazivac na podatke za C  
28     const int ldc  
29     //Velicina prve dimenzije za C  
30 )
```

Upotreba

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <blas.h>
4
5  int main() {
6      double *A, *B, *C;
7      int m = 3;
8      int i, j;
9      A = (double*)malloc(m*m*sizeof(double));
10     B = (double*)malloc(m*m*sizeof(double));
11     C = (double*)malloc(m*m*sizeof(double));
12     for(i=0;i<m;i++){
13         for(j=0;j<m;j++){
14             A[i + m*i] = j + m*i;
15             B[j + m*i] = 3.14*(j+m*i);
16             C[j + m*i] = 0.0;
17         }
18     }
```

Upotreba

```
19     double alpha = 1.0;
20     double beta = 0.0;
21     cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans,
22     m,m,m,alpha,A,m,B,m,beta,C,m
23     );
24     for(i = 0;i < m;i++){
25         for(j = 0;j<m;j++){
26             printf("A[%d][%d]=%g ",i, j, A[j+m*i]);
27         }
28         printf("\n");
29     }
30     for(i = 0;i < m;i++){
31         for(j = 0;j<m;j++){
32             printf("B[%d][%d]=%g ",i, j, B[j+m*i]);
33         }
34         printf("\n");
35     }
```


Upotreba

```
36     for(i = 0; i < m; i++) {  
37         for(j = 0; j < m; j++) {  
38             printf("C[%d][%d]=%g ", i, j, C[j+m*i]);  
39         }  
40         printf("\n");  
41     }  
42  
43     free(A); free(B); free(C);  
44     return 0;  
45 }
```

BLAS primena

- ▶ BLAS se intenzivno koristi u pozadini velikog broja paketa baziranih na linearnoj algebri.
- ▶ Direktno se i ne koristi puno, ne više.
- ▶ Korisno je savladati kako radi zato što se koncepti iz BLAS sveta pojavljuju na dosta mesta.

Lapack

- ▶ Linear Algebra PACKage.
- ▶ Naslanja se na BLAS i zamenjuje stari Linpack.
- ▶ Lapack je implementiran u Fortran-u.
- ▶ Verzija za C (koja je deo Lapack projekta) je Lapacke.
- ▶ Sve rutine počinju sa LAPACKE_ prefiksom koji prati kod koji se sastoji od:
 - ▶ Tipa podataka (kao u BLAS)
 - ▶ Tipa matrice (kao u BLAS uz par dodatnih)
 - ▶ Operacije

LAPACK operacije

Ime	Opis
SV	Rešava sisteme linearnih jednačina.
LS, LSY, LSS, LSD	Rešava problem najmanjih kvadrata, minimizuje x u formuli ravnoj drugoj normi razlike između vektora b i matrice A puta vektor x .
LSE	Kao gore, ali sa ograničenjem da Bx jednako d .
GLM	Rešenje problema opšteg linearnog modela, tj. minimiziranje x bazirano na drugoj normi vektora y sa ograničenjem $d = Ax + By$
EV, EVD,EVR	Računanje sopstvene vrednosti i sopstvenih vektora simetrične matrice A

LAPACK operacije

Ime	Opis
ES	Traži sopstvene vrednosti i vektore za nesimetrične matrice A
SVD, SDD	Dekompozicija matrice.

Primer LAPACK funkcije

```
1 lapack_int LAPACKE_dgesv(  
2     int matrix_layout,  
3     //LAPACK_ROW_MAJOR ili LAPACK_COL_MAJOR  
4     lapack_int n,  
5     //broj jednacina  
6     lapack_int nrhs,  
7     //broj vrednosti sa desne strane jednacine  
8     double* a,  
9     //nxn matrica A  
10    lapack_int lda,
```

Primer LAPACK funkcije

```
11 //prva dimenzija niza a
12 lapack_int* ipiv,
13 //niz duzine n u koji ce biti spremljena
14 //permutacija matrice tokom faktORIZACIJE
15 double* b,
16 //nxnhrs matrica desnih strana, kasnije tu budu resenja
17 lapack_int ldb
18 //prva dimenyija niza b
19 );
```

Upotreba

```
1  #include <stdio.h>
2  #include <lapacke.h>
3
4  int main(int argc, const char** argv){
5      double A[3][3] = {1, 3, 2, 4, 1, 9, 5, 7, 2};
6      double B[3] = {-1, -1, 1};
7      lapack_int ipiv[3];
8      lapack_int info, m, lda, ldb, nrhs;
9      int i, j;
10     m=3;
11     nrhs=1;
12     lda=3;
13     ldb=1;
```


Upotreba

```
14 info=LAPACKE_dgesv(LAPACK_ROW_MAJOR, m, nrhs, *A,lda,ipiv,b,ldb);
15 if(info > 0){
16     printf("U(%d,%d) je nula. A je singularna matrica.\n",info,info);
17     return 0;
18 }
19 for(i=0;i<m;i++){
20     printf("b[%i] = %g\n",i,b[i]);
21 }
22 printf("\n");
23 return 0;
24 }
```

Skalabilnost BLAS/LAPACK

- ▶ Podrazumevano, sve rutine u BLAS/LAPACK su serijske.
- ▶ Rešenje? Postoje HPC-optimizovane verzije koje rade iste stvari, ali imaju MPI sposobnosti:
 - ▶ ScaLapack
 - ▶ PBLAS

Alternative

- ▶ GNU Scientific Library (GSL)
 - ▶ *Izuzetno* široka biblioteka.
 - ▶ Ima ponešto za svaku temu.
 - ▶ Za nas je najzanimljivije što ima BLAS interfejs koji čini bazične operacije sa matricama nešto manje mučnim.
- ▶ SuperLU
 - ▶ Supernodal LU
 - ▶ Fokusira se na rešavanje ogromih sistema jednačina.
 - ▶ Ima izuzetnu podršku za ubrzanje kroz OpenMP, MPI, i čak GPU ubrzanje

Alternative

▶ PETSc

- ▶ Portable extensible toolkit for scientific computation
- ▶ Uglavnom namenjen za rešavanje diferencijalnih jednačina, ali kao deo posla rešavanja sistema parcijalnih diferencijalnih jednačina ima odlične sisteme za rešavanje običnih linearnih jednačina.
- ▶ Posедуje HPC podršku direktno u samoj biblioteci.
- ▶ Koristi MPI
- ▶ Intenzivno se koristi u HPC krugovima

▶ SLEPc

- ▶ Scalable Library for Eigenvalue Problem Computations
- ▶ Proširenje PETSc koje računa i probleme sopstvenih vrednosti i vektora

Alternative

▶ ELPA

- ▶ Eigenvalue Solvers for Petaflop-Applications
- ▶ Još jedan način da se na HPC-optimizovan način računaju sopstvene vrednosti i vektori.
- ▶ Ova varijanta je naročito popularna u kvantnoj hemiji i nauci materijala zbog svoje sposobnosti da brzo računa sopstvene vektore Hermitijanskih matrica.

▶ HYPRE

- ▶ Kombinuje se sa PETSc i proširuje njene sposobnosti da rešava sisteme linearnih jednačina.
- ▶ Koristi MPI

Ostale primene

Parcijalne diferencijalne jednačine

- ▶ PETSc
 - ▶ Već je pomenut.
 - ▶ Specijalizovan je za baš diferencijalne jednačine.
 - ▶ Podržava i rešavanje nelinearnih jednačina.
- ▶ Trilinos projekat
 - ▶ Veliki sistem za brze naučne proračune napravljen da radi na velikom broju HPC arhitektura.

Grafovi

- ▶ PBGL
 - ▶ Parallel Boost Graph Library
 - ▶ Bazira se na Boost-ovoj biblioteci za grafove ali je proširuje da radi u HPC okruženju.
 - ▶ Radi dobro sa arhitekturama distribuirane memorije.
- ▶ Combinatorial BLAS
 - ▶ Oslanja se na i formiran je po ugledu na BLAS
 - ▶ Takođe cilja na arhitekture sa distribuiranom memorijom
- ▶ Giraph

I/O

- ▶ NetCDF
 - ▶ Format razvijen za geologiju, geografiju, i geofiziku.
 - ▶ Karakteriše ga odlična podrška za arhiviranje.
- ▶ HDF5
 - ▶ Sa ovim, kako čujem, imate dosta iskustva.
- ▶ Silo
 - ▶ Biblioteka višeg nivoa koja se oslanja na HDF5 da olakša unos iz ispisivanje podataka u distribuiranim aplikacijama.

Dekompozicija meš struktura

- ▶ METIS

- ▶ Tretira meš kao graf odn. hipergraf i dekomponuje na osnovu toga.
- ▶ Serijalna je, ali ima HPC-optimizovanu verziju: ParMETIS koja koristi MPI

- ▶ Trilinos

- ▶ Trilinos podržava i ovu funkcionalnost.

Vizuelizacija

- ▶ O ovome, celo predavanje, ali najbitnija biblioteka u ovoj oblasti jeste VTK: Visualization Toolkit.