Rachunek macierzowy

Maciej Paszyński

Katedra Informatyki
Akademia Górniczo-Hutnicza w Krakowie
home.agh.edu.pl/paszynsk

Materiały do ćwiczeń

Wprowadzenie

1. Wykład składa się z dwóch części: Rachunek macierzowy oraz Statystyka wielowymiarowa
2. Zaliczenie części z Rachunku Macierzowego polegać będzie na napisaniu kilku programów w Octavie oraz na rozwiązaniu kilku zadań pisemnych.
3. Na początku każdego wykładu od 28 lutego do 3 kwietnia będę zadawał do rozwiązania jedno zadanie. Na końcu każdego wykładu będę zbierał rozwiązania zadanych wcześniej zadań. Ostatni raz zadanie zostanie zadane 3 kwietnia. Zadania można oddać maksymalnie pod koniec wykładu mającego miejsce 3 tygodnie od zadania.
4. Zadania oceniane będą zero-jedynkowo, w sumie można więc zebrać 6 punktów. Konieczne jest zapisanie obliczeń.
5. Proszę przynieść na ćwiczenia laptopy z zainstalowanym Octave. Na ćwiczeniach będziemy pisać programy w grupach 2-3 osobowych. Niektóre programy będą używały poprzednich. Zadania z wykładu będą pomagać zrozumieć programy z ćwiczeń. Wszystko będzie powiązane w pewną całość.

Wprowadzenie

1. Na każdych ćwiczeniach będzie zadawany prosty program do napisania w grupie 2-3 osobowej, oraz będzie możliwość pokazania napisanych wcześniej programów. W sumie zadanych zostanie 6 programów od 28 lutego do 3 kwietnia. Program można oddać maksymalnie do 3 tygodni po ich zadaniu, na ćwiczeniach mających miejsce 3 tygodnie po zadaniu programu.
2. Oceny z programów będą zero-jedynkowe. Program albo daje poprawne wyniki albo nie, z dokładnością do trzeciego miejsca po przecinku.
3. Ocena $= MIN\{(Programy \in [0-6] + Zadania \in [0-6])/2; 5\}$
4. Zadania zadawane na wykładach nie są obowiązkowe. Możliwe jest uzyskanie zaliczenia z tej części punktów poprzez napisanie jednego dużego (90 minut) kolokwium zaliczeniowego pisemnego na przełomie kwietnia / maja.
5. Konsultacje: wtorki od 16.30 do 18.00 po uprzednim umówieniu e-mailowym

Programy (przykład: mnożenie macierzy)

MANUAL: *home.agh.edu.pl/paszynsk/RM/MATLAB.pdf*
W programach w MATLABie (Octavie) używamy tylko podstawowych operacji.

```
function [C] = MM(A,B)
% Przykladowe uzycie:  A=rand(100,100); B=rand(100,100); C=MM(A,B);
% tic; C=MM(A,B); toc %zmierz czas i wypisz czas
sza = size(A);
szb = size(B);
if(sza(2)!=szb(1))
  disp('Nie odpowiednie rozmiary macierzy');
  return
endif
C(1:sza(2),1:szb(1))=0;
for i=1:sza(2)
  for j=1:szb(1)
    for k=1:sza(1)
      C(i,j)+=A(i,k)*B(k,j);
    endfor
  endfor
endfor
endfunction
```

Programy (przykład: blokowe mnożenie macierzy)

```
function [C] = MMblock(A,B)
sza = size(A); szb = size(B); n=sza(1); nhalf = sza(1)/2;
if(sza(1)==2)
  C(1,1)=A(1,1)*B(1,1)+A(1,2)*B(2,1);
  C(1,2)=A(1,1)*B(1,2)+A(1,2)*B(2,2);
  C(2,1)=A(2,1)*B(1,1)+A(2,2)*B(2,1);
  C(2,2)=A(2,1)*B(1,2)+A(2,2)*B(2,2);
  return
endif
C(1:nhalf,1:nhalf)=...
  MMblock(A(1:nhalf,1:nhalf),B(1:nhalf,1:nhalf))+...
  MMblock(A(1:nhalf,nhalf+1:n),B(nhalf+1:n,1:nhalf));
C(1:nhalf,nhalf+1:n)=...
  MMblock(A(1:nhalf,1:nhalf),B(1:nhalf,nhalf+1:n))+...
  MMblock(A(1:nhalf,nhalf+1:n),B(nhalf+1:n,nhalf+1:n));
C(nhalf+1:n,1:nhalf)=...
  MMblock(A(nhalf+1:n,1:nhalf),B(1:nhalf,1:nhalf))+...
  MMblock(A(nhalf+1:n,nhalf+1:n),B(nhalf+1:n,1:nhalf));
C(nhalf+1:n,nhalf+1:n)=...
  MMblock(A(nhalf+1:n,1:nhalf),B(1:nhalf,nhalf+1:n))+...
  MMblock(A(nhalf+1:n,nhalf+1:n),B(nhalf+1:n,nhalf+1:n));
endfunction
```

Programy

Poprawność działania programu sprawdzamy porównując z wbudowaną funkcjonalnością MATLABa (Octave),
np. wynik powyższej procedury mnożenia macierzy z wynikiem obliczonym używając funkcjonalności MATLABa

```
» A=[1 2; 3 4];
» MM(A,A)
ans =
7 10
15 22
» A*A
ans =
7 10
15 22
» MMblock(A,A)
ans =
7 10
15 22
```

(opuszczenie średnika powoduje wypisanie wyniku)

Pierwszy zestaw deadline 20 marca

Implementacja w Octave w grupach 2-3 osobowych różnych wersji mnożenia macierzy

1. Mnożenie macierzy jako złożenie mnożenia macierzy przez wiele wektorów (wykład 1 slajd 13) dla macierzy 3x3
2. Mnożenie macierzy jako złożenie mnożenia wierszy przez kolumny (wykład 1 slajd 17) dla macierzy 3x3
3. Mnożenie macierzy jako suma wielu rank-1 update (wykład 1 slajd 19) dla macierzy 3x3
4. Mnożenie macierzy (n parzyste) jak blokowe mnożenie wielu macierzy 2x2 (wykład 1 slajd 34) dla macierzy 3x3
5. Rekurencyjne mnożenie macierzy z wykorzystaniem algorytmu Strassena (wykład 1 slajd 44) dla macierzy 4x4

Drugi zestaw deadline 3 kwietnia

Implementacja w Octave w grupach 2-3 osobowych różnych wersji algorytmu Gaussa i LU faktoryzacji

1. Rozwiązanie układu równań metodą Chińska z 179 r p.n.e. (wykład 2 slajd 3)
2. Algorytm eliminacji Gaussa generujący jedynki na przekątnej macierzy bez pivotingu (wykład 2 slajd 15)
3. Algorytm eliminacji Gaussa generujący jedynki na przekątnej macierzy z pivotingien (wykład 2 slajd 21)
4. Algorytm eliminacji Gaussa bez pivotingu "generujący wyznacznik" na przekątnej macierzy U (wykład 2 slajd 24 lub 28)
5. Algorytm eliminacji Gaussa z pivotingiem (wykład 2 slajd 30)
6. Algorytm LU faktoryzacji (wykład 2 slajd 37)

Trzeci zestaw deadline 10 kwietnia

Implementacja w Octave w grupach 2-3 osobowych algorytmów obliczających różne normy i wskaźniki uwarunkowania macierzy

1. Algorytm obliczania normy Frobeniusa $\|A\|_F$ (wyklad 3 slajd 7)
2. Algorytm obliczania normy $\|A\|_1$ (wyklad 3 slajd 9-11)
3. Algorytm obliczania normy $\|A\|_\infty$ (wyklad 3 slajd 12)
4. Algorytm obliczania wyznacznika macierzy $\|A\|$ (wyklad 3 slajd 19-21)
5. Algorytm obliczania macierzy A^{-1} (wyklad 3 slajd 22)
6. Algorytm obliczania współczynnika uwarunkowania macierzy $cond_1(A)$ (wyklad 3 slajd 31)
7. Algorytm obliczania wspołczynnika uwarunkowania macierzy $cond_\infty(A)$ (wyklad 3 slajd 32)
8. Algorytm obliczania wspołczynnika uwarunkowania macierzy $cond_F(A)$ (wykład 3 slajd 30)

Czwarty zestaw deadline 17 kwietnia

1. Algorytm obliczania wartości własnych macierzy $\|A\|$ dla macierzy 2x2 (wykład 4 slajd 16)
2. Metoda potęgowa obliczania wartości własnych macierzy $\|A\|$ dla macierzy 2x2 (wykład 4 slajd 23)
3. Metoda potęgowa obliczania wartości własnych macierzy $\|A\|$ dla macierzy 3x3 (wykład 4 slajd 23)
4. Metoda potęgowa obliczania wartości własnych macierzy $\|A\|$ dla macierzy 4x4 (wykład 4 slajd 23)
5. Algorytm obliczania normy $\|A\|_2$ dla macierzy 2x2 (wykład 4 slajd 8)
6. Algorytm obliczania normy $\|A\|_2$ dla macierzy 3x3 (wykład 4 slajd 6)
7. Algorytm obliczania normy $\|A\|_2$ dla macierzy 4x4 (wykład 4 slajd 6)

Piąty zestaw deadline 24 kwietnia

1. Algorytm obliczania singular value decomposition (SVD) dla macierzy $\|A\|$ dla macierzy 2x2
2. Algorytm obliczania singular value decomposition (SVD) dla macierzy $\|A\|$ dla macierzy 2x3
3. Algorytm obliczania singular value decomposition (SVD) dla macierzy $\|A\|$ dla macierzy 3x2
4. Algorytm obliczania singular value decomposition (SVD) dla macierzy $\|A\|$ dla macierzy 3x3

Szósty zestaw deadline 24 kwietnia

1. Dla zadanego wektora z $\mathcal{R}^2$, tworzenie macierzy odbicia Q i przemnażanie wektora tak żeby wyzerować jego drugą współrzędną (wykład 6 slajd 28-30)
2. Dla zadanego wektora z $\mathcal{R}^3$,tworzenie macierzy odbicia Q i przemnażanie wektora tak żeby wyzerować jego drugą współrzędną (wykłąd 6 slajdy 28-30)
3. Dla zadanego wektora z $\mathcal{R}^3$,tworzenie macierzy odbicia Q i przemnażanie wektora tak żeby wyzerować jego trzecią współrzędną (wykład 6 slajdy 28-30)
4. Dla zadanej macierzy 2x2 QR faktoryzacja przez odbicia (wykład 6 slajdy 29-30)
5. Dla zadanej macierzy 3x3 QR faktoryzacja przez obroty (wykład 6 slajdy 29-30)