

# ToDo

*Przemysław Chojewski*

*17 11 2019*

## zadanie

1. **(2 osoby)** Rozwiązywanie układu równań liniowych  $Ax = b$  z wykorzystaniem blokowej metody Crouta, gdzie  $A(n \times n)$  jest macierzą postaci

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{12}^T & A_{11} & A_{23} \\ 0 & A_{23}^T & A_{11} \end{pmatrix},$$

gdzie  $A_{ij}(p \times p)$  i  $n = 3p$ . Zakładamy, że  $A_{11}$  jest macierzą ortogonalną. Do rozwiązywania odpowiednich układów równań liniowych zastosować metodę eliminacji Gaussa z częściowym wyborem elementu głównego (GEPP).

## Co zostało już zrobione?

1. Wstępny algorytm podziału  $A = L * U$
2. GEPP:
  - a) GEPPp, czyli  $A * x = b$
  - b) GEPPM, czyli  $A * X = M$  (poza rozwiązaniem  $U * X = M\_nowa$ )
  - c) blok3, czyli  $L * x = b$ , gdzie  $L$  - blokowa specyficzna
3. losoweA generujący macierze z zadania

## Co jest do zrobienia w najbliższym czasie?

1. Sprawdzić, czy może w blok3 założyć, że  $L_{11}$  jest ortogonalna.
2. Przeczytać notatki zapisane w pliku CroutPartition.m i zobaczyć, czy jest tam coś wartościowego.
3.  $U * X = M\_nowa$  w GEPPM.

## Co jest do zrobienia?

- 0) Zastanowić się jakie są głębsze konsekwencje tego, że  $A_{11}$  jest ortogonalne.

- 1) Testy błędów względnych, współczynników stabilności oraz współczynnika poprawności.
- 2) Porównanie czasu działania algorytmu ze standardowym matlabowym  $A \setminus b$ :
  - a) sprawdzenie, jak działa `tic toc`, czy można dodawać wartości?
- 3) Znaleźnięcie za pomocą generatora ciekawych macierzy (pewnie `hilb`, czy `cos`)
- 4) Opisanie działania algorytmu w *LaTeX*.
- 5) Wykresy:
  - a) prędkości naszego algorytmu-porównanie z  $A \setminus b$ .

NOTE: Być może `losoweA` jest wystarczająco stabilne czasowo dla stałych  $n$ , żeby korzystać z niego w pętli i wykonywać obliczenia na bieżąco. Trzeba to sprawdzić. Jeśli czasem zajmuje mu zrobienie macierzy 9999 4 sekundy, a czasem 7, to nie można w pętli, ale jeżeli wariancja jest nieduża, to można.
  - b) błędy względne