Pierwszy projekt z labu

Zamiast w octave można zaprogramować odpowiednią funkcję i testy w Pascalu, C/C++, fortranie.

Projekt składa się z dwóch części:

1. napisać funkcję octave'a GSR() z metodą znajdowania rozkładu QR A wymiaru $M \times N$ kolumnami regularnej tzn $A = Q * R$ dla
   (a) Q wymiaru $M \times N$ takiej, że $Q^T Q = I$ - kolumny są ortonormalne
   (b) R górnotrójkątnej nieosobliwej wymiaru $N \times N$ z dodatnią główną przekątną

   blokowym rekurencyjnym algorytmem Gramm-Schmidta tzn. jeśli macierz ma 1 kolumne ($n = 1$) - to $R = \|A\|_2$ (A to wektor a R to macierz 1×1 czyli skalar) i wtedy $Q = A/R$; w przeciwnym przypadku w kolejnych krokach liczymy:

   (a) dzielimy blokowo kolumny macierzy A tzn A=[A1,A2] np. tak by ilość kolumn obu podmacierzy była równa - lub prawie równa (choć algorytm działa dla dowolnego podziału np biorąc za $A1$ pierwszą kolumne A)
   (b) wywołujemy funkcję [Q1,R1]=GSR(A1) znajdując odpowiedni rozkład QR tzn. $Q1 * R1 = A1$,
   (c) ortogonalizujemy kolumny $A2$ względem układu ortonormalnego kolumn $Q1$ tzn. licząc $B = A2 - Q1 * R12$ (jak dobrać $R12$? To zazwyczaj nie jest macierz trójkątna...) tak by kolumny B były ortogonalne do kolumn $Q1$ tzn. by $B^T * Q1 = 0$ (to oczywiście wektor zerowy),
   (d) znaleźć rozkład QR macierzy $B = Q2 * R2$ znów wywołując GSR,
   (e) tworzymy nasze macierze rozkładu QR tzn. Q=[Q1,Q2] i R=[R1,R12;0,R2] - zero oznacza blok zerowy odpowiedniego wymiaru.

   Zauważmy, że zachodzi równość A1=Q1*R1=[Q1,Q2]*[R1;0] a także A2=Q1*R12+B=Q1*R12+Q2*R2=[Q1,Q2]*[R12;R2]. (wszędzie używamy składni octave'a do zapisywania macierzy blokowego, przecinki oddzielają bloki wierszowo, średniki kolumnowo).

   Parametrem funkcji:

```
function [Q,R]=GSR(A)
% kod funkcji
end
```

ma być: macierz A,

Funkcja ma zwracać macierze Q, R czyli czynniki rozkładu QR Q o kolumnach ortonormalych i R górnotrójkątna o dodatniej przekątnej.

Funkcja powinna sprawdzać bez dodatkowych obliczeń czy macierz jest kolumnami regularna tzn. zwracać ostrzeżenie kiedy nie jest maksymalnego rzędu BEZ wykorzystania jakiejś kosztownej funkcji typu qr(), svd(), rank() etc.

Oczywiście na jakimś poziomie tolerancji tzn. obliczane elementy, które są zerowe dla A NIE o max rzędzie będą w arytmetyce fl tylko 'bliskie' zero więc jeśli będą poniżej pewnej tolerancji np. $1e-12$ przyjmiemy że macierz numerycznie osobliwa - jak to zrobić to zadanie dla Państwa.

2. **Testy**:

   (a) **Test czy metoda działa** Przetestować na kilku prostych przykładach dla losowych macierzy $m \times n$ z $m \geq n$ - np. $m = n+1$ czy $m = n+10$ dla $n = 3, 10, 20$ sprawdzając czy $\|Q * R - A\|_1$ małe i czy kolumny Q są rzeczywiście ortonormalne (lub bliskie ortonormalności) - górnotrójkątność mamy zapewnioną (dlaczego?) i nie musimy jej sprawdzać.

   (b) **Test dla trudnych przykładów** wziąć złą numerycznie macierz np. pierwszych n kolumn macierzy Hilberta $n+1 \times n+1$ dla różnych n i powtórzyć testy jak w poprzednim punkcie.

   (c) **Zastosowanie do LZNK** Zastosować tę funkcję do rozwiązania LZNK z losową macierzą A 21×20 i losowym wektorem prawej strony f tzn liczymy $b = Q^T f$ i rozwiązujemy $Rx = b$ - porównać z wynikiem otrzymanym operatorem *backslash*.