Laboratorium projektowe Metod Numerycznych -- zadanie *rozgrzewkowe C*

Metody iteracyjne dla układów równań liniowych

Należy napisać funkcję o wywołaniu postaci

```
x = itlinsolc(ciA,vicA,b)
```

przybliżającą rozwiązanie układu równań liniowych $\boldsymbol{A}\,x = b$, gdzie $\boldsymbol{A}$ jest nieosobliwą macierzą rzeczywistą o dominującej (wierszowo) głównej przekątnej. O macierzy $\boldsymbol{A}$ wiadomo, że w każdym wierszu ma dokładnie d niezerowych elementów, gdzie d = **`size(ciA,2)`**. Rozmiar macierzy $\boldsymbol{A}$ jest równy n, gdzie n = **`size(ciA,1)`**, a pozycje i wartości jej niezerowych elementów zapisane są w macierzach **`ciA`** i **`vicA`**. Macierz **`ciA`** zawiera w i-tym wierszu numery pozycji (numery kolumn) niezerowych elementów i-tego wiersza macierzy $\boldsymbol{A}$, a macierz **`vicA`** odpowiadające im wartości. Dodatkowo wiadomo, że $2 \le d \le 50$ oraz $2 \le n \le 5000000$. Można przyjąć, że elementy w wierszach macierzy **`ciA`** są uporządkowane rosnąco.

Wyznaczone przybliżenie x rozwiązania układu $\boldsymbol{A}\,x = b$ nazwiemy wystarczająco dokładnym, jeśli $\|b - \boldsymbol{A}\,x\|_1 \le 10^{-9}$, przy czym wartość 10^{-9} nie jest traktowana bardzo restrykcyjnie (oznacza to, że można liczyć w napisanym programie jakieś w miarę rozsądne oszacowanie/przybliżenie tego błędu, a nie sam błąd). *W ładnym programie wartość* 10^{-9} *powinna być nazwana!*

Ważne: koniecznie trzeba zadbać o to, aby funkcja nie działała zbyt długo. W szczególności nie należy wykonywać więcej niż 500 iteracji wybranej metody iteracyjnej. Jeśli do tego czasu nie uda się wyznaczyć odpowiednio dokładnego przybliżenia rozwiązania rozważanego układu równań liniowych, należy przypisać do wyniku x macierz pustą i zakończyć program. *W ładnym programie wartość 500 powinna być nazwana!*

Nazwy argumentów funkcji **`itlinsolc`** mogą być dowolne, zachowana musi być tylko składnia oraz nazwa funkcji. Nie trzeba sprawdzać poprawności argumentów wejściowych.

Rozwiązując to zadanie można użyć dowolnej metody numerycznej -- nie musi to być nawet metoda iteracyjna, ale być może żadna z metod *dokładnych* nie zadziała w praktyce w wypadku tak dużych rozmiarów macierzy.

Rozwiązaniem zadania rozgrzewkowego jest tylko i wyłącznie program realizujący to zadanie, czyli funkcja (plik) **`itlinsolc.m`** oraz ewentualne funkcje (pliki **`*.m`**) pomocnicze.

Zasady oceniania

Oceniana będzie poprawność działania funkcji, jej efektywność oraz estetyka (i modularyzacja) kodu. Za poprawne i nie za wolne rozwiązanie tego zadania można uzyskać do **4** punktów. Za rozwiązanie poprawne i bardzo szybkie do **6** punktów. Autorzy wyróżniających się rozwiązań tego zadania mogą otrzymać bonusowe punkty (maksymalnie **2** punkty).

Uwagi

Zadania ***C*** i ***R*** są identyczne poza sposobem przechowywania i rozkładem niezerowych elementów macierzy A.

Przykład macierzy **`ciA`** i **`vicA`** ($n = 9$, $d = 3$):

```
ciA =
     1     5     7
     1     2     8
     3     6     9
     4     7     8
     1     5     8
     3     5     6
     6     7     9
     7     8     9
     2     4     9

vicA =
    1.5247    0.2571    1.0205
    0.8617   -4.7229    0.9298
   -0.6349    0.0012    0.2398
    0.9894   -0.0708   -0.6904
   -0.6516    7.2773   -2.4863
    1.1921    0.5812   -5.2240
   -1.6118    9.8936   -2.1924
   -0.0245    4.4709   -2.3193
   -1.9488    0.0799    3.7832
```

Wtedy:

```
A =
    1.5247         0         0         0    0.2571         0    1.0205         0         0
    0.8617   -4.7229         0         0         0         0         0    0.9298         0
         0         0   -0.6349         0         0    0.0012         0         0    0.2398
         0         0         0    0.9894         0         0   -0.0708   -0.6904         0
   -0.6516         0         0         0    7.2773         0         0   -2.4863         0
         0         0    1.1921         0    0.5812   -5.2240         0         0         0
         0         0         0         0         0   -1.6118    9.8936         0   -2.1924
         0         0         0         0         0         0   -0.0245    4.4709   -2.3193
         0   -1.9488         0    0.0799         0         0         0         0    3.7832
```