Grzegorz Fotyga
Piotr Kowalczyk
Rafał Lech
Krzysztof Nyka
Piotr Sypek

9 marca 2023

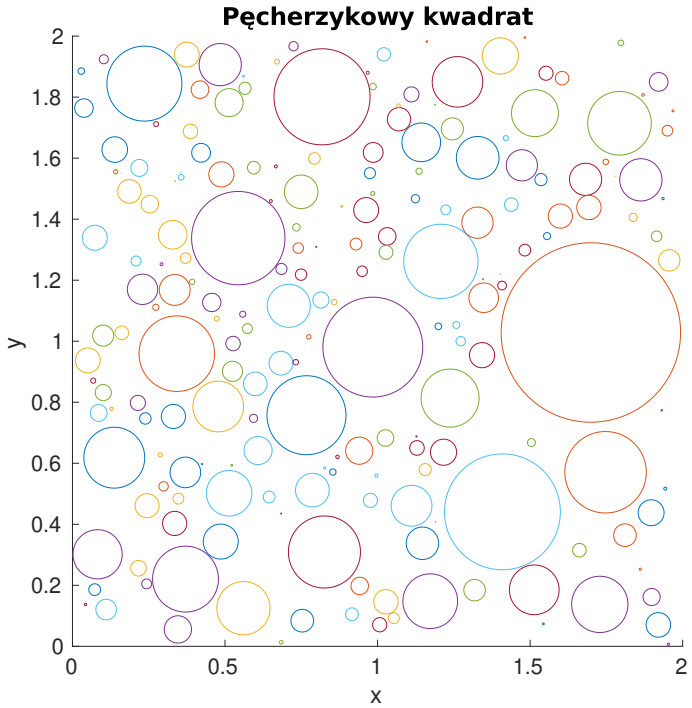
Metody numeryczne, laboratorium 2.

W ramach drugiego laboratorium z Metod numerycznych należy zrealizować dwa zadania. W celu zaliczenia tego laboratorium należy zaliczyć oba zadania.

1. Zadanie 1. Pęcherzyki

Celem pierwszego zadania jest opracowanie kodu, który do kwadratu o boku `a` wpisywać będzie nieprzecinające się okręgi (pęcherzyki). Położenie okręgów powinno być wybierane losowo poprzez wywołanie funkcji **`rand`**. Promienie okręgów powinny być losowane z zakresu (0, `r_max`]. Okrąg powinien być rysowany tylko jeśli będzie mieścił się w kwadracie oraz nie będzie przecinał się z innymi okręgami. Żaden okrąg nie powinien zawierać innego okręgu. Liczba okręgów określona przez zmienną `n_max` powina wynosić 200.

Przykładowy rezultat końcowy programu *Pęcherzyki* przedstawia rys. 1.



Rysunek 1: Przykład okręgów wygenerowanych przez program *Pęcherzyki*

Do narysowania okręgu o promieniu `R`, którego środek znajduje się w punkcie (`X`,`Y`) można zastosować funkcję z wydruku 1. Zawartość wydruku 1 można skopiować i wkleić do pliku *plot_circle.m*.

```
function plot_circle(X, Y, R)
    theta = linspace(0,2*pi);
    x = R*cos(theta) + X;
    y = R*sin(theta) + Y;
    plot(x,y)
end
```

Wydruk 1: Funkcja do rysowania okręgu

Program *Pęcherzyki* można opracować w następujących etapach:

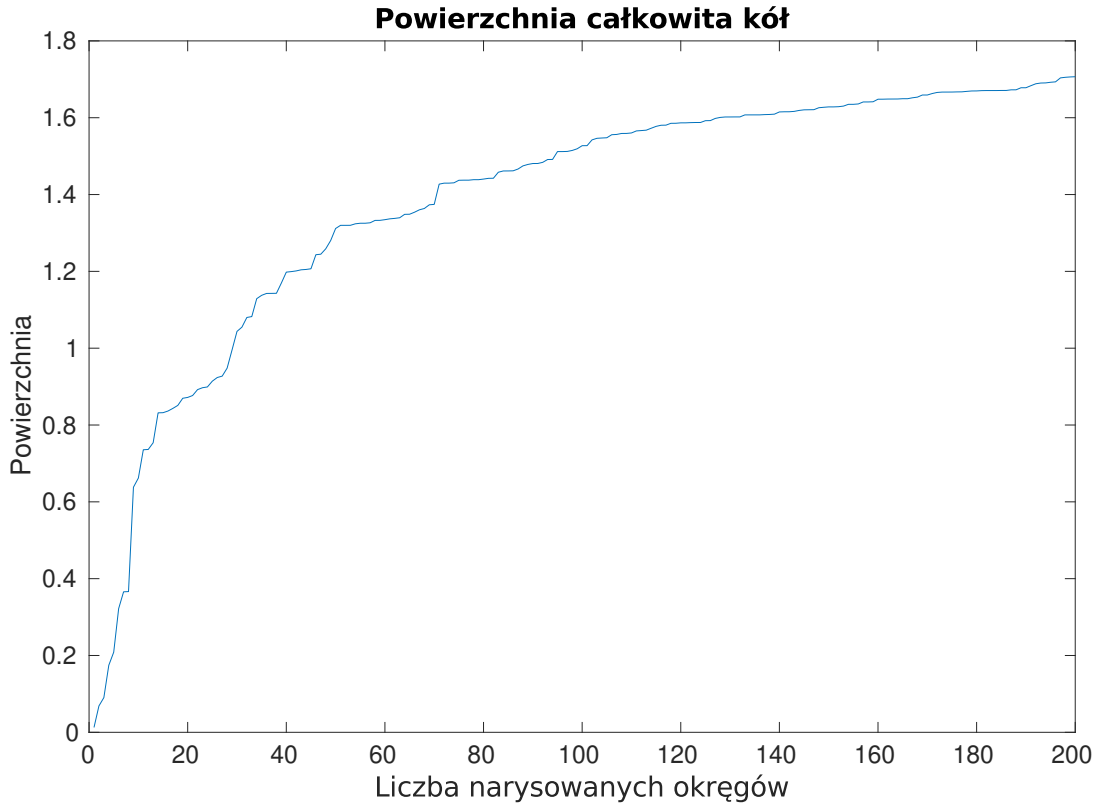
1. W skrypcie *zadanie1.m* zdefiniuj parametry początkowe: dowolne `a`, `r_max=a/2`, `n_max=200`. W dwóch pierwszych linijkach kodu wywołaj polecenia
   - **`clear all`** w celu usunięcia wszystkich zmiennych wygenerowanych przez poprzednie wywołanie skryptu,
   - **`close all`** w celu zamknięcia wszystkich okien z wykresami.
2. Utwórz i narysuj pierwszy okrąg z losowo określonym położeniem i promieniem. Do losowania wartości parametrów okręgu użyć funkcji **`rand`** i skalowania do wartości maksymalnej parametru, czyli do `a` lub `r_max`. Uwaga: sprawdź w linii poleceń rezultat działania komend **`rand`**`(1)` oraz **`rand`**`(5)`. Po pierwszym wywołaniu funkcji rysującej okrąg należy dodać dwie komendy:
   - **`axis`** `equal` – wyrównanie skali osi x i y,
   - **`axis`**`([0 a 0 a])` – ograniczenie wyświetlanego obszaru do $x \in [0, a]$ oraz $y \in [0, a]$.
3. Zastosuj pętlę **`while`** do wylosowania i narysowania `n_max` okręgów. Do zliczania liczby narysowanych okręgów zastosuj zmienną `n`. Po narysowaniu pierwszego okręgu należy wywołać komendę **`hold`** `on`, która spowoduje, że elementy dotychczas dodane do wykresu zostaną zachowane po kolejnym wywołaniu polecenia **`plot`**. Dodatkowo, po każdym narysowaniu okręgu powinna być wywoływana komenda **`pause`**`(0.01)`.
4. Dodaj warunek, który będzie sprawiał, że rysowane okręgi będą zawarte w kwadracie o boku `a`. Uwaga: upewnij się, że liczba narysowanych okręgów wynosi `n_max`.
5. Zapamiętaj losowane wartości parametrów okręgów:
   (a) zapamiętaj parametry `n`–tego okręgu w wektorach `x`, `y`, `r`, np. `x(n)=X`,
   (b) zapamiętaj powierzchnię każdego wylosowanego okręgu (nazwa zmiennej dowolna),
   (c) zapamiętaj liczbę losowań przeprowadzonych w celu narysowania każdego z wylosowanych okręgów (nazwa zmiennej dowolna).
6. Dodaj warunek, który będzie sprawiał, że rysowane okręgi nie będą się przecinać. Do zdefiniowania tego warunku należy m.in. odczytywać wartości wektorów `x`, `y`, `r`.

1.1. Zadania do wykonania

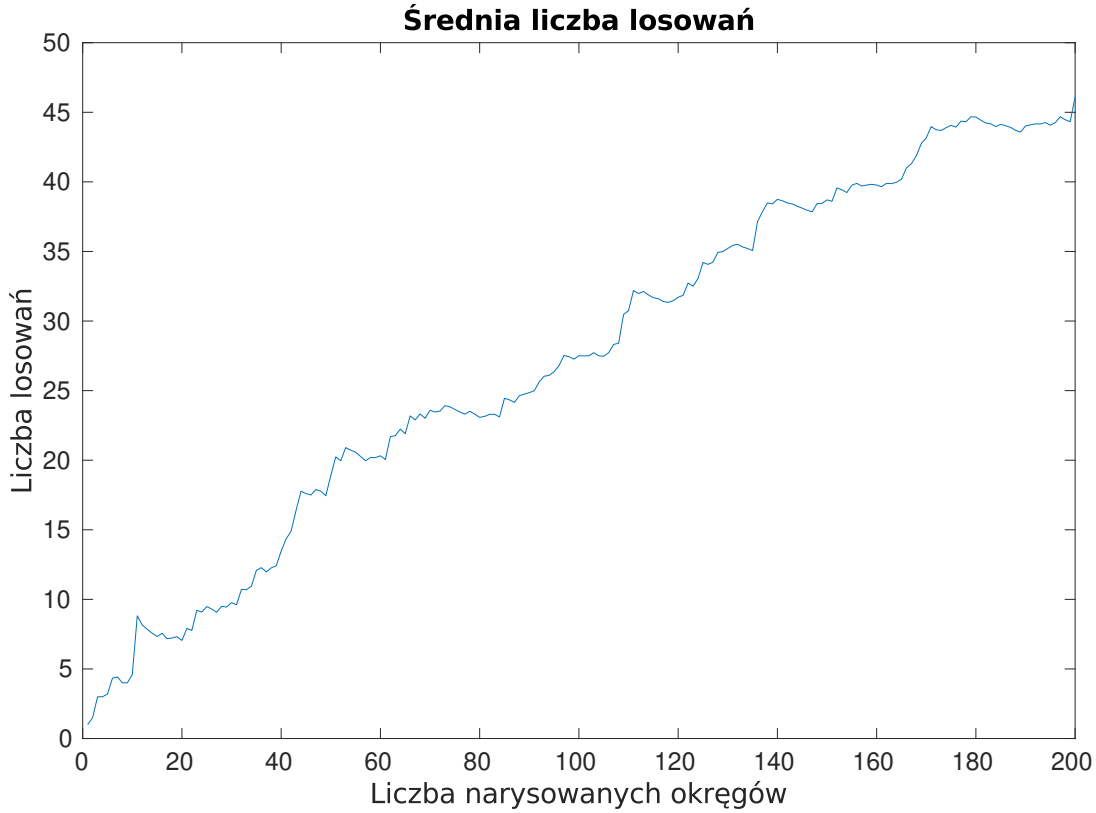
Do zaliczenia pierwszego zadania wymagane jest napisanie programu *Pęcherzyki*. Dodatkowe dwa zadania są wymagane do otrzymania maksymalnej liczby punktów:

- **Zadanie A** – narysuj wykres, który będzie przedstawiał sumę powierzchni kół zawartych w rysowanych okręgach w funkcji liczby dotychczas narysowanych okręgów. Przykład wykresu przedstawia rys. 2. Do wyznaczenia sumy powierzchni kół można zastosować funkcję **`cumsum`**, która wyznacza sumę skumulowaną (ang. *cumulative sum, prefix sum*). Uwaga: nowe okno wykresu można otworzyć poprzez wywołanie **`figure`**.
- **Zadanie B** – narysuj wykres przedstawiający historię średniej liczby losowań wymaganych do narysowania pierwszych `n` okręgów. Do przygotowania tego wykresu można zastosować funkcję **`cumsum`**, jednak należy pamiętać, że we wzorze na średnią występuje dzielenie. Przykład wykresu przedstawia rys. 3.

Uwaga: wykresy powinny zawierać tytuł oraz opis osi (polecenia **`xlabel`**, **`ylabel`**, **`title`**).



Rysunek 2: Powierzchnia całkowita zajmowana przez n pierwszych wylosowanych kół



Rysunek 3: Średnia liczba losowań wymagana do narysowania n pierwszych okręgów

2. Zadanie 2. PageRank

2.1. Wstęp

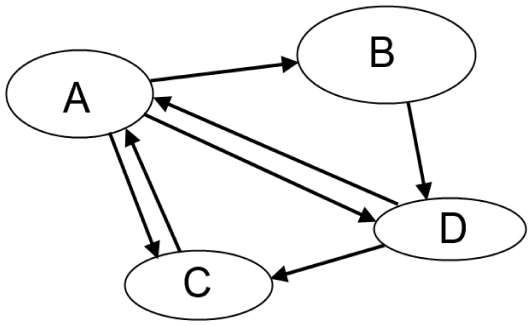
Drugie zadanie polega na opracowaniu implementacji podstawowej wersji algorytmu PageRank. Algorytm PageRank służy do numerycznej oceny jakości stron internetowych w sieci komputerowej na podstawie relacji między poszczególnymi stronami. Rozszerzona wersja tego algorytmu stanowiła podstawę sukcesu firmy Google.

Uwagi:

- W oryginalnej wersji algorytmu PageRank tworzone jest tzw. równanie własne, które jest rozwiązywane za pomocą metody potęgowej. W ramach niniejszego laboratorium zostanie omówiony wariant PageRank, który operuje na układzie równań liniowych. Algorytm ten dokładniej opisano w [1].
- Na tym laboratorium przedstawione zostały podstawowe założenia algorytmu PageRank. Szczegóły implementacyjne tego algorytmu, które zostały opracowane przez firmę Google, nie zostały upublicznione i nadal stanowią tajemnicą firmy Google.
- W opracowanej na tym laboratorium implementacji algorytmu PageRank **należy zastosować w obliczeniach macierze rzadkie**, których cechy opisano w p. 2.5.
- Kod z zadania drugiego będzie wymagany do wykonania zadań na trzecim laboratorium.

2.2. PageRank – ogólne założenia

Algorytm PageRank przypisuje każdej ze stron internetowych pewien współczynnik (o nazwie takiej samej jak nazwa algorytmu – PageRank, oznaczony w skrócie przez PR), który określa ważność tej strony względem innych stron. Współczynnik ten jest związany z prawdopodobieństwem, że internauta klikając w sposób losowy w odnośnik, ostatecznie trafi na rozpatrywaną stronę.



Rysunek 4: Sieć składająca się z czterech stron internetowych (A, B, C i D)

Rozpatrzmy prostą sieć składającą się z czterech stron internetowych (A, B, C i D), przedstawioną na rys. 4. Połączenia między stronami można przedstawić w formie macierzy `Edges`, która dla sieci z rys. 4 przybiera w Matlabie postać:

```
Edges = [ A, A, A, B, C, D, D;
          B, C, D, D, A, A, C ];
```

gdzie górny wiersz zawiera strony z odnośnikami do stron z dolnego wiersza. Przykładowo, do strony D odnoszą się dwie strony: A i B. Ranking strony D, oznaczony jako $PR(D)$, zdefiniowano w (1).

$$PR(D) = \frac{PR(A)}{3} + \frac{PR(B)}{1} \tag{1}$$

$PR(D)$ jest równy sumie rankingów wszystkich stron, które posiadają odnośniki do D, podzielonych przez liczbę odnośników wychodzących z tych stron (trzy odnośniki wychodzące z A i jeden odnośnik wychodzący z B).

2.3. PageRank – konstrukcja liniowego układu równań

W wyniku rozpisania (1) dla wszystkich N stron w sieci internetowej powstaje układ równań liniowych, który można zapisać jako następujące liniowe równanie macierzowe:

$$\boldsymbol{M}\boldsymbol{r} = \boldsymbol{b} \tag{2}$$

- Poszukiwany wektor $\boldsymbol{r}$ zawiera wartości PR wszystkich N stron w sieci.
- Prawa strona równania, wektor $\boldsymbol{b}$, ma długość N. Wszystkie elementy wektora $\boldsymbol{b}$ mają wartość $(1-d)/N$, gdzie d jest współczynnikiem tłumienia.
- Macierz $\boldsymbol{M}$ jest rzadka i ma postać:

$$\boldsymbol{M} = \boldsymbol{I} - d\boldsymbol{B}\boldsymbol{A} \tag{3}$$

  - $\boldsymbol{I}$ jest macierzą jednostkową (w Matlabie: `I = speye(N)`).
  - $\boldsymbol{B}$ jest macierzą sąsiedztwa:

$$\boldsymbol{B} = \begin{bmatrix} b(1,1) & b(1,2) & \cdots & b(1,N) \\ b(2,1) & b(2,2) & \cdots & b(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ b(N,1) & b(N,2) & \cdots & b(N,N) \end{bmatrix} \tag{4}$$

    gdzie $b(i,j) = 1$, jeżeli strona j ma odnośnik do strony i (**zwróć uwagę na kolejność indeksów**), $b(i,j) = 0$ w pozostałych przypadkach.
  - $\boldsymbol{A}$ jest macierzą diagonalną:

$$\boldsymbol{A} = \begin{bmatrix} 1/L(1) & 0 & \cdots & 0 \\ 0 & 1/L(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/L(N) \end{bmatrix} \tag{5}$$

    gdzie $L(j)$ jest liczbą odnośników wychodzących z j–tej strony, którą można wyznaczyć korzystając z macierzy $\boldsymbol{B}$:

$$L(j) = \sum_{i=1}^{N} B(i,j) \tag{6}$$

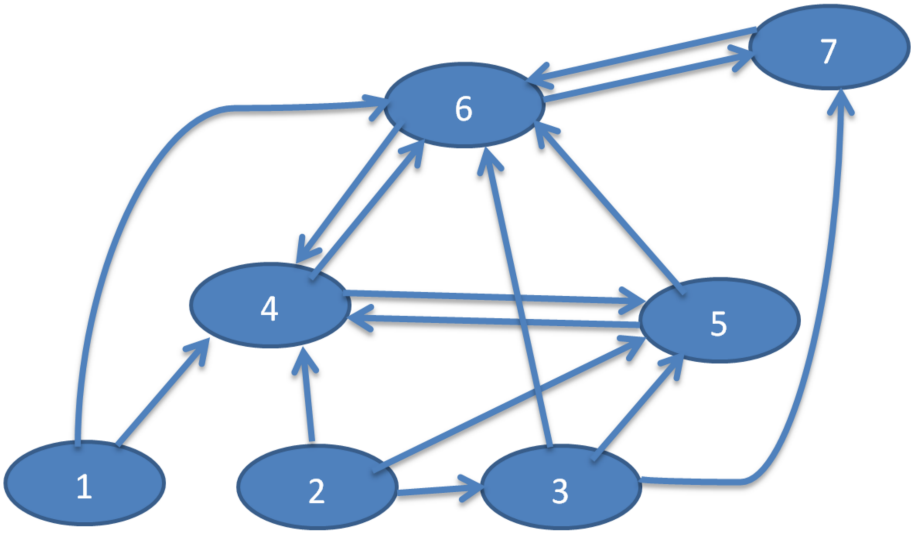
W wyniku rozwiązania liniowego równania macierzowego (2) otrzymujemy wektor $\boldsymbol{r}$, który zawiera wartości PR wszystkich N stron w sieci. Im wyższa wartość PR tym strona jest bardziej istotna w rozpatrywanej sieci.

Szczegółowy opis przekształceń prowadzących do wyznaczenia (2) przedstawiono w [1].

2.4. Zadania do wykonania

W trakcie zajęć zapoznaj się przede wszystkim z poleceniami: **sparse**, **speye**, **spdiags**, **diag**.

- **Zadanie A** – zdefiniuj tablicę `Edges` dla sieci złożonej z siedmiu stron przedstawionej na rys. 5.
- **Zadanie B** – skonstruuj macierze $\boldsymbol{B}$, $\boldsymbol{A}$, $\boldsymbol{I}$, $\boldsymbol{M}$ oraz wektor $\boldsymbol{b}$ dla połączeń wygenerowanych w Zadaniu A oraz dla parametru $d = 0.85$. Macierze $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{I}$, $\boldsymbol{M}$ powinny być macierzami rzadkimi, natomiast wszystkie wektory powinny być macierzami gęstymi z jedną kolumną. Do utworzenia macierzy $\boldsymbol{B}$ wywołaj funkcję **sparse** z pięcioma argumentami: jako pierwszy argument podaj indeksy wierszy elementów niezerowych macierzy $\boldsymbol{B}$, czyli drugi wiersz macierzy `Edges` pobrany przez wywołanie `Edges(2,:)`. Drugim argumentem funkcji **sparse** powinien być wektor indeksów kolumn macierzy $\boldsymbol{B}$. Trzeci argument funkcji **sparse** określa wartości elementów niezerowych, które dla macierzy $\boldsymbol{B}$ są równe 1, czyli trzeba jako trzeci argument przesłać wektor zawierający same jedynki. Macierz zawierającą tylko jedynki generuje funkcja `ones`. Do utworzenia macierzy $\boldsymbol{I}$ zastosuj funkcję **speye**. Wektor $\boldsymbol{L}$ stanowi sumę elementów kolejnych kolumn macierzy $\boldsymbol{B}$, którą można otrzymać poprzez wywołanie **sum**`(B)`. Macierz $\boldsymbol{A}$ zbuduj poprzez wywołanie funkcji **spdiags**.



Rysunek 5: Sieć składająca się z siedmiu stron

- **Zadanie C** – wykonaj polecenie `whos A B I M b` i umieść rezultat tego polecenia w pliku sparse_test.txt, który należy dołączyć do sprawozdania. Za to zadanie przyznane będzie 0,25 punktu, jeśli wszystkie macierze będą macierzami rzadkimi oprócz wektora `b`, który powinien stanowić macierz gęstą (w metodach numerycznych wektor oznacza najczęściej macierz z jedną kolumną, znacznie rzadziej macierz z jednym wierszem).
- **Zadanie D** – wykonaj polecenie `spy(B)` i zapisz wygenerowany wykres do pliku png stosując polecenie `print -dpng spy_b`. Upewnij się, czy odpowiednie elementy macierzy $\boldsymbol{B}$ są niezerowe.
- **Zadanie E** – rozwiąż układ równań (2) za pomocą metody bezpośredniej (`r = M\b`).
- **Zadanie F** – zaobserwuj wartość PageRank dla poszczególnych stron w sieci poprzez wygenerowanie wykresu poleceniem `bar(r)`. Wygenerowany wykres zapisz do pliku o nazwie *bar.png*.

Sprawozdanie do zadania 2 powinno zawierać kody oraz dwa wykresy i jeden plik txt.

2.5. Macierze rzadkie

W metodach numerycznych macierz może zostać zapisana jako macierz gęsta (ang. dense matrix, full storage) lub macierz rzadka (ang. **sparse matrix**). **Macierz gęsta** jest to macierz, której każdy element posiada wartość zapisaną w pamięci. **Macierz rzadka** to macierz, która opisana jest przez strukturę danych o rozmiarze proporcjonalnym do liczby elementów niezerowych tej macierzy. Macierz rzadka stosowana jest do reprezentacji macierzy zawierających znaczną liczbę zerowych elementów. Jest to głównie spowodowane tym, że dla takich macierzy macierz rzadka wymaga znacznie mniej pamięci niż macierz gęsta. Dodatkowo, istnieje wiele algorytmów dedykowanych do działań na macierzach rzadkich, które silnie redukują liczbę operacji zmiennoprzecinkowych potrzebnych do ich wykonania w porównaniu do tych samych obliczeń realizowanych na macierzach gęstych.

Istnieje wiele formatów macierzy rzadkich. Jednym z nich jest format COO, który przechowuje informację tylko o elementach niezerowych macierzy w trzech tablicach: (1) tablica indeksów wierszy, (2) tablica indeksów kolumn, (3) tablica wartości elementów niezerowych macierzy. Przykład zapisu macierzy w formacie COO został przedstawiony na stronie software.intel.com.

Polecam obejrzenie blisko pięciominutowego filmu z konferencji C++Now, który przybliża znaczenie macierzy rzadkich: https://www.youtube.com/watch?v=t1cYMEv-zr0.

[1] http://www.cse.unt.edu/ tarau/teaching/NLP/PageRank.pdf