Techniki Obliczeniowe w Nauce i Technice
Układy równań liniowych i dekompozycje macierzy

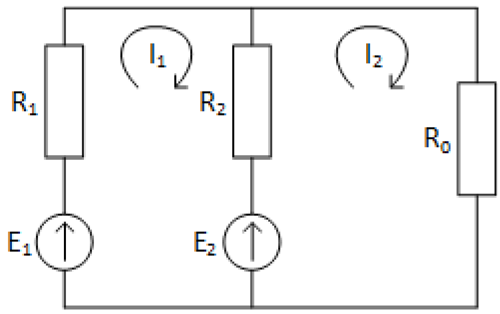
dr inż. Jacek Wszołek (na podstawie prof. Tomasz Zieliński)

2016.10.22

1. Metoda podstawień Gaussa-Jordana(2 pkt)

Korzystając z II prawa Kirchhoffa dokonaj macierzowego zapisu równań oczkowych poniższego obwodu elektrycznego obrazującego połączenie równoległe akumulatorów. Przyjmij, że rezystancje wewnętrzne akumulatorów **R_1** oraz **R_2** wynoszą odpowiednio **5** i **1** [mΩ], rezystancja rozrusznika **R_0=0.1[Ω]**, natomiast wartości napięć wynoszą: **E_1 = 12,1** [V] oraz **E_2=11,9** [V]. Na tej podstawie wypełnij odpowiednimi wartościami macierze **A** i **b** z równania:

$$\mathbf{A}\,\mathbf{x}=\mathbf{b}$$



Następnie metodą Gaussa-Jordana wyznacz analitycznie (na papierze) rozwiązanie napisanego przez siebie układu równań oraz **napisz program w Matlabie**, który rozwiązuje tą metodą dowolne równanie z N niewiadomymi, a nie tak jak w niniejszym przykładnie, tylko z dwoma.

2. Metoda LU (2 pkt)

Napisz program w Matlabie rozwiązujący równanie z zadania (1) metodą dekompozycji LU. Sprawdź czy wyniki otrzymane z wykorzystaniem obu metod są takie same. Następnie skonstruuj układ 100 równań ze 100 niewiadomymi (w formie macierzowej) i za pomocą poleceń `tic toc` lub funkcji MATLABa `cputime()` porównaj czas działania programu z programem z ćwiczenia 1. Dane do układu równań wylosuj z wykorzystaniem funkcji `randi()`. W celu zwiększenia dokładności pomiaru czasu możesz w pętli wykonywać wiele prób rozwiązywania (różnych!) układów równań a otrzymany czas podzielić przez liczbę rozwiązanych układów równań.

3. Metody iteracyjne (1 pkt)

Zaimplementuj w MATLABie jedną z iteracyjnych metod rozwiązywania układów równań. Następnie wykorzystaj ją do rozwiązania układu równań z macierzą rzadką. Do generacji macierzy rzadkiej możesz wykorzystać skrypt `macierz_rzadka.m`. Podobnie jak w zadaniu 2 porównaj czas rozwiązywania układów równań z wykorzystaniem wybranej przez siebie metody iteracyjnej z metodą LU.

4. Syntezator mowy (opcjonalnie 2 pkt)

Kodery mowy, wykorzystywane w telefonii cyfrowej, wykorzystują znany mechanizm (znany model) generacji mowy. Na podstawie fragmentu sygnału mowy wyznacza się wartości parametrów modelu generacji, kwantuje się je, a następnie przesyła z telefonu mówcy do telefonu odbiorcy. Następnie w telefonie odbiorcy syntezuje się mowę, wykorzystując model generacji (syntezy) mowy. W tym przykładzie przeanalizujemy i zsyntezujemy samogłoskę „a". Musimy zidentyfikować parametry filtru traktu głosowego oraz jego pobudzenie, odpowiadające głosce „a".

1) Nagraj 3 sekundy jakiejś samogłoski za pomocą karty dźwiękowej z częstotliwością próbkowania 8000 Hz (czyli 8000 próbek na sekundę). Lub skorzystaj z załączonego pliku `A.wav`. Otrzymasz sygnał $x(n)$, gdzie $n=0,1,2,\dots,N-1$ (N to liczba wszystkich próbek/liczb).

```
fpr=8000; x = wavrecord(3*fpr, fpr); lub [x,fpr]=audioread(‘A.wav’);
plot(x); soundsc(x, fpr);
```

2) Wyznacz pierwszych $K=128$ współczynników funkcji autokorelacji sygnału $x(n)$, czyli wartości $r(0), r(1), r(2), \ldots, r(K-1)$:

$$\mathbf{r}(k)=\sum_{n=k}^{N-1} x(n) x(n-k) \qquad k=1,2,3,\ldots,N-1$$

Narysuj $r(k)$ i znajdź wartość indeksu k, dla którego funkcja osiąga maksimum (nie uwzględniaj okolicy $k=0$). Oznacz ten indeks przez T (np. Otrzymasz $T=70$) - jest to okres otwierania strun głosowych (okres impulsowego sygnału, pobudzającego filtr traktu głosowego);

3) Skonstruuj macierz **R** i wektor **r** oraz oblicz wektor współczynników **a** według poniższych wzorów (dla $P=10$):

$$\mathbf{R}=\begin{bmatrix} r(0) & r(1) & \cdots & r(P-1) \\ r(1) & r(2) & \cdots & r(P-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & \cdots & r(0) \end{bmatrix}, \quad \mathbf{r}=\begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(P) \end{bmatrix} \quad \rightarrow \quad \mathbf{a}=\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{bmatrix}=-R \backslash r$$

Współczynniki **„a”** są współczynnikami filtru traktu głosowego.

4) Zsyntezuj $M=24000$ próbki sygnału $y(n)$ według wzoru, czyli pobudzaj w określony sposób filtr traktu głosowego:

$$y(n)=e(n)-\sum_{k=1}^{P} a_k y(n-k), \qquad n=0,1,2,\ldots,N-1$$

zakładając $y(-1)=y(-2)=y(-3)=\ldots=0$ oraz przyjmując, że w sygnale $e(n)$ co T próbek występuje „1” (otwarcie strun głosowych) , a poza tym składa się on z samych zer (struny głosowe są zamknięte).

5) Narysuj sygnał $y(n)$ i odsłuchaj go na karcie dźwiękowej:

```
plot(y); soundsc(y,fpr);
```