Metoda sprzężonych gradientów dla macierzy wstęgowej

Tomasz Chwiej

12 marca 2018

Zadanie polega na rozwiązaniu układu równań liniowych $A\boldsymbol{x} = \boldsymbol{b}$ metodą sprzężonych gradientów.

1 Zadania do wykonania:

1. Utworzyć macierz układu o wymiarze $n = 1000$ i wypełnić jej elementy zgodnie z poniższą formułą:

$$\begin{aligned} A[i][j] &= \frac{1}{1+|i-j|}, \quad gdy\ |i-j| \leqslant m, \quad i,j = 0,\ldots,n-1 \\ A[i][j] &= 0, \quad gdy\ |i-j| > m \end{aligned}$$

   Przyjąć $m = 5$.

2. Utworzyć wektor wyrazów wolnych b. Jego elementy wypełnić następująco:

$$b[i] = i + 1, \quad i = 0,\ldots,n-1 \tag{1}$$

3. Utworzyć wektor startowy x. Jego elementy wypełnić następująco:

$$x[i] = 0, \quad i = 0,\ldots,n-1 \tag{2}$$

4. Zaprogramować metodę sprzężonego gradientu do rozwiązania układu równań liniowych. Proszę użyć zmodyfikowanego algorytmu (wektor mnożymy przez macierz tylko raz w danej iteracji):

$$\begin{aligned} &//inicjalizacja \\ \boldsymbol{v}_1 &= \boldsymbol{r}_1 = \boldsymbol{b} - A\boldsymbol{x}_1 \\ - &\quad - \quad - - - - - - - \\ &//petla\ iteracyjna\ CG \\ &while(\boldsymbol{r}_k^T\boldsymbol{r}_k > 10^{-6})\{ \\ \alpha_k &= \frac{\boldsymbol{r}_k^T\boldsymbol{r}_k}{\boldsymbol{v}_k A\boldsymbol{v}_k} \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k\boldsymbol{v}_k \\ \boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k A\boldsymbol{v}_k \\ \beta_k &= \frac{\boldsymbol{r}_{k+1}^T\boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T\boldsymbol{r}_k} \\ \boldsymbol{v}_{k+1} &= \boldsymbol{r}_{k+1} + \beta_k\boldsymbol{v}_k \\ &\} \end{aligned}$$

5. Rozwiązać zdefiniowany powyżej układ równań przy użyciu metody sprzężonych gradientów. W każdej iteracji należy zapisać do pliku: aktualny numer iteracji (k), wartość normy euklidesowej wektora reszt ($\|\boldsymbol{r}_k\|_2 = \sqrt{\boldsymbol{r}_k^T \boldsymbol{r}_k}$), wartość α_k, wartość β_k, wartość normy euklidesowej wektora rozwiązań ($\|\boldsymbol{x}_k\|_2 = \sqrt{\boldsymbol{x}_k^T \boldsymbol{x}_k}$). Jako warunek zakończenia przyjąć że: $\sqrt{\boldsymbol{r}_k^T \boldsymbol{r}_k} < 10^{-6}$.

6. Sporządzić wykresy: $\|\boldsymbol{r}_k\|_2 = f(k)$ oraz $\|\boldsymbol{x}_k\|_2 = f(k)$, gdzie: k - numer iteracji. Dla $\|\boldsymbol{r}_k\|_2$ wprowadzić skalę logarytmiczną (w gnuplocie: **set logscale y**).

7. W domu proszę rozwiązać powyższy układ równań $A\boldsymbol{x} = \boldsymbol{b}$ metodą eliminacji zupełnej.

8. W sprawozdaniu proszę przeanalizować rozwiązanie oraz porównać wydajności obu metod (sprzężonych gradientów i eliminacji zupełnej) - wydajniejsza metoda działa oczywiście szybciej. Z czego wynika tak duża różnica w wydajności? Odpowiedź proszę uzasadnić bazując na liczbie wykonywanych operacji. **Dla chętnych: Jaki czas jest potrzebny na rozwiązanie układu przy użyciu obu metod, gdy $n = 10^4$? Co z zajętością pamięci (macierz układu)?**

2 Uwagi praktyczne:

- funkcję max(x,y) można zdefiniować jako makro

```
#define max(X,Y) ((X)>(Y)? (X):(Y))
```

- funkcję min(x,y) można zdefiniować jako makro

```
#define min(X,Y) ((X)<(Y)? (X):(Y))
```

- funkcję abs(i-j) można zdefinować jako makro

```
#define abs(X)  ((X)>0? (X):-(X))
```

- Aby wyznaczyć czas wykonania części kodu należy: a) dołączyć plik nagłówkowy **time.h**, b) użyć dwukrotnie funkcji **time(time_t *t)**, która zwraca aktualny czas, c) różnicę dwóch czasów t_2 i t_1 wyznaczyć przy użyciu funkcji **difftime(t2,t1)**. W skrócie wyglądałoby to tak:

```
#include <time.h>
int main(){

 time_t t1,t2;
 double t21;

 time(&t1);             //start
 gaussj(a,n,x,1);
 time(&t2);             //koniec
 t21=difftime(t2,t1);   //roznica daje czas wykonania
}
```

- Mnożenie $\boldsymbol{y} = A\boldsymbol{x}$ w przypadku symetrycznej macierzy wstęgowej o liczbie $2m + 1$ przękątnych można zrealizować następująco:

```
for(i=0;i<n;i++){
   jmin=max(0,i-m);
   jmax=min(i+m,n-1);
   y[i]=0;
   for(j=jmin;j<=jmax;j++)y[i]+=A[i][j]*x[j];
}
```