Sprawozdanie z numerycznych metod rozwiązywania cząstkowych równań różniczkowych

Karol Urbański

15 czerwca 2011

1 Podstawy teoretyczne

1.1 Rozwiązywany problem

Rozwiązywany problem to zadanie 1 z laboratorium, w którym znajdujemy rozwiązanie *równania Laplace'a* w przestrzeni dwuwymiarowej $(x, y) \in [0, 1]^2$:

$$\frac{\delta^2 V}{\delta x^2} + \frac{\delta^2 V}{\delta y^2} = 0 \tag{1}$$

z następującymi warunkami początkowymi:

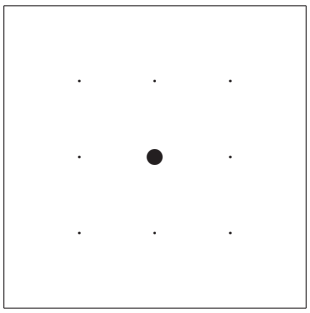
$$\forall x, y \in R : V(x, 0) = V(x, 1) = V(0, y) = V(1, y) = 0,$$

$$\forall (x, y) \in C : V(x, y) = l,$$

gdzie C to kwadratowy obszar przewodnika taki, że $C \in [0, 1]^2$, a l to dana stała oznaczająca potencjał przewodnika.

1.2 Dyskretyzacja problemu

Przestrzeń, w której rozwiązujemy równanie, musimy udyskretyzować, aby móc wykorzystać metody numeryczne do przybliżenia rozwiązania problemu. Dyskretyzacji dokonujemy poprzez podział przestrzeni na siatkę $(n-1)^2$ punktów $p_{ij} \in [0, 1]^2$ takich, że $\forall (i, j) \in \{1, 2, ..., n-1\} \times \{1, 2, ..., n-1\} : p_{ij} = (ih, jh), h = \frac{1}{n}$.



Rysunek 1: Przykładowa dyskretyzacja dla $n = 4$, punkty oznaczone kropkami, punkty należące do przewodnika w centrum (pogrubiony); obwódka oznacza otaczający ekran

Ponieważ do rozwiązania problemu będziemy wykorzystywać MRS[1] dla równania stacjonarnego, dla ułatwienia obliczeń musimy zastosować dyskretyzację w taki sposób, by zmapować siatkę dwuwymiarową do jednowymiarowej. Wykonujemy ją w najprostszy możliwy sposób:

$$p_k = p_{ij} \iff k = i \cdot (n-1) + j$$

czyli poprzez przypisanie kolejnych indeksów kolejnym elementom w kolejnych wierszach (idąc od góry do dołu oraz od lewej do prawej). Przypisanie to jest bardzo proste, ale niezbędne jest zadbanie o dokładne przygotowanie obliczeń, aby uniknąć błędów.

1.3 Przybliżanie pochodnych drugiego stopnia

W MRS korzystamy z zastąpienia pochodnych ilorazem różnicowym korzystającym do obliczenia przybliżonej wartości pochodnej przy użyciu punktów siatki znajdujących się w bezpośrednim otoczeniu badanego punktu. W przedstawionym problemie korzystamy z drugiej różnicy centralnej dla kroku h

$$\frac{\delta^2 f}{\delta x^2} \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} \tag{2}$$

obliczającej przybliżoną wartość drugiej pochodnej.[2]

[1] MRS - metoda różnic skończonych

[2] Obliczona wartość jest dobrym przybliżeniem wartości drugiej pochodnej - z dokładnością do wyrazów drugiego rzędu ($O(h^2)$)

1.3.1 Zastosowanie ilorazów różnicowych do zadanego problemu

Dla każdego z punktów siatki zastępujemy pochodne ilorazem różnicowym. Używamy kroku $h = \frac{1}{n}$ oraz wcześniej wprowadzonej notacji punktów p_{ij}.

$$\frac{\delta^2 V}{\delta x^2} = \frac{V(x_{i-1j}) + V(x_{i+1j}) - 2V(x_{ij})}{h^2} \tag{3}$$

$$\frac{\delta^2 V}{\delta y^2} = \frac{V(x_{ij-1}) + V(x_{ij+1}) - 2V(x_{ij})}{h^2} \tag{4}$$

Podstawiając (3) i (4) do (1) otrzymujemy przybliżone rozwiązanie

$$\frac{V(x_{i-1j} + V(x_{ij-1}) - 4V(x_{ij}) + V(x_{ij+1}) + V(x_{i+1j})}{h^2} = 0 \tag{5}$$

co możemy zapisać przy użyciu naszego mapowania jako

$$\frac{V(x_{k-n}) + V(x_{k-1}) - 4V(x_k) + V(x_{k+1}) + V(x_{k+n})}{h^2} = 0 \tag{6}$$

Jest to układ $(n-1)^2$ równań z tylomaż niewiadomymi $V(x_k)$. Ma on postać

$$AV = b \tag{7}$$

gdzie A w jednowymiarowej wersji MRS byłoby dla naszego rozwiązania macierzą piecioprzekątniową o wartościach $\frac{-4}{h^2}$ na głównej przekątnej oraz $\frac{1}{h^2}$ na przekątnej n wierszy poniżej, 1 wiersz poniżej, 1 wiersz powyżej oraz n wierszy powyżej.[3] Takie ustawienie powodowałoby jednak, że nasze rozwiązanie brałoby wtedy na brzegach obszaru pod uwagę wartości z pól o jedno pole w lewo bądź prawo - które nie sąsiadują z danym elementem. Dlatego, w macierzy A, w wierszach $k \in in, i \in 1, 2, .., n-1$ zastępujemy wartość ponad główną przekątną zerem. Podobnie, w następnym wierszu zastępujemy wartość pod przekątną. Ponieważ warunek brzegowy (ekran) w problemie ma potencjał równy 0, wektor wyrazów wolnych ma (na razie) postać

$$b = [0, 0, ..., 0]^T$$

1.3.2 Uwzględnienie przewodnika wewnątrz obszaru

Musimy jeszcze uwzględnić źródło potencjału - przewodnik. By to zrobić, w wektorze wyrazów wolnych wiersze odpowiadające punktom siatki należącym do przewodnika zastępujemy wartością $\frac{l}{h^2}$. Następnie, by upewnić się, że wartość potencjału w przewodniku po obliczeniach będzie dalej równa l oraz niezależna od wartości w okolicznych polach, każdy wiersz macierzy A o indeksie i takim, jak punkt siatki należący do przewodnika zastępujemy wierszem, w którym $A_{ii} = 1$, a każdy pozostały element wiersza i jest równy 0. Wtedy $V(x_i)$ będzie miał zawsze wartość l po zakończeniu obliczeń, czyli pożądaną przez nas sytuację. Dodatkowo, nie musimy uwzględniać żadnych dodatkowych warunków w wektorze wolnym - rozwiązanie równania zrobi to za nas.

[3] Warto zauważyć, że (6) ma postać pozwalającą na skrócenie h^2, co znacznie uprości warstwę implementacyjną, ale jest bez znaczenia dla teoretycznych rozważań.

1.4 Rozwiązanie równania

Tak przygotowany układ równań (7) musimy rozwiązać, by otrzymać nasz rozkład potencjału. Do rozwiązywania nie użyjemy metody eliminacji Gaussa (ani innej metody bezpośredniej), gdyż ma ona wiele wad. Najgorszą z nich jest ogromna złożoność dla wystarczająco dużych macierzy - biorąc pod uwagę, że siatka 50x50 ma 2500 punktów, złożoność $O(n^3)$ jest absolutnie nie do zaakceptowania dla naszego problemu, którego ilość zmiennych rośnie kwadratowo wraz ze zmniejszeniem kroku. Żeby uniknąć problemu gigantycznej złożoności użyjemy algorytmu iteracyjnego: metody Gaussa-Seidla. Metoda ta w każdej sukcesywnej iteracji ulepsza przybliżenie z poprzedniej iteracji dla układu $Ax = b$, zaczynając od dowolnie wybranego x_0. Metoda ta bazuje na dekompozycji macierzy A i działa w miejscu, co jest dodatkowym atutem metody. Jest zbieżna dla układów, w których dominują elementy na kilku przekątnych (czyli jak w naszym problemie). W samym algorytmie wykorzystany zostanie wzór roboczy o następującej postaci:

$$x_i^{(t+1)} = \frac{1}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(t+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(t)}] \tag{8}$$

Jak widać, pierwsza z sum jest liczona na podstawie wcześniej obliczonych elementów w tej samej iteracji - metoda działa w miejscu, czyli ma znacznie mniejszą złożoność pamięciową.

Implementacja algorytmu znajduje się w warstwie implementacyjnej (rozdział 2.1.2).

2 Warstwa implementacyjna

2.1 Kernel obliczeniowy

2.1.1 Wykonanie

Kernel obliczeniowy jest to program w C, przyjmujący na wejściu następujące argumenty:

1. wartość początkowa l
2. ilość punktów siatki wzdłuż jednego boku
3. współrzędna x lewego górnego rogu punktów siatki należących do przewodnika
4. współrzędna y lewego górnego rogu punktów siatki należących do przewodnika
5. współrzędna x prawego dolnego rogu punktów siatki należących do przewodnika

6. współrzędna y prawego dolnego rogu punktów siatki należących do przewodnika

2.1.2 Rozwiązywanie układu równań

Do rozwiązania układu równań wykorzystany jest następujący algorytm. Działa do uzyskania odpowiednio małego błędu obliczeń.

```
void gauss_seidel(float** mat,float* x,float* vec,int size_sqr){
        int i,j,k=0;
        float prev;
        float sum;
        float maxerr=0.01;
        while(maxerr>0.0001){
                maxerr=0.0;
                for(i=0; i<size_sqr; i++){
                        prev=x[i];
                        sum=0.0;
                        for(j=0;j<i;j++)
                                sum+=mat[i][j]*x[j];
                        for(j=i+1;j<size_sqr;j++)
                                sum+=mat[i][j]*x[j];
                        x[i]=(vec[i]-sum)/mat[i][i];
                        if(fabs(x[i]-prev)>maxerr)
                                maxerr=fabs(x[i]-prev);
                }
                k++;
        }
}
```

2.2 Komunikacja z interfejsem graficznym

Interfejs graficzny otrzymuje dane poprzez strumień danych bezpośrednio z programu. Możliwe jest również użycie prekalkulowanych wartości podanych na wejściu interfejsu. Pierwsza linia danych podana do interfejsu to typ dawanych danych (CMODE jeżeli wywołujemy program w C, PREMODE jeżeli wykorzystujemy stare obliczenia). Następna linia w CMODE to argumenty wywołania programu w C, a w PREMODE gotowa do wyświetlenia tablica punktów.

2.3 Interfejs

Interfejs napisany jest w perlu, z użyciem OpenGL. Wyświetla płytkę z odpowiednim gradientem kolorów (niebieski dla dodatnich potencjałów, czarny dla zerowych, zielony dla ujemnych). Może wyświetlać w trybie wireframe (klawisz w), trybie punktowym (klawisz m), oraz z wypukłościami dla wartości różnych od zera (wyświetlanie w 3D, klawisz p).

3 Pomiary, obserwacje

3.1 Testy czasu wykonania

Do pomiaru czasu wykonania użyłem skryptu shella (zsh):

```
for i in {1 .. 10}; do
        /usr/bin/time -f '%e' ./calc 1 n 5 5 10 10
done
```

ilość punktów na brzegu	$\bar{t}$	σ
10	0.00s	0.00s
25	0.127s	0.010s
50	3.412s	0.023s
100	79.662s	0.784s