Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

Algorytmy i Metody Optymalizacji

Projekt 1, zestaw nr. 49

Dominika Ziółkiewicz

rok akademicki 2024/2025

Contents

1 Wstęp

Celem projektu jest wyznaczenie naszego położenie na kuli ziemskiej na podstawie danych na temat położenia pięciu satelitów oraz czasów propagacji pochodzących od nich sygnałów. Oprócz sformułowania i rozwiązania zadania optymalizacji, zostanie również przedstawiony wpływ zmian określonych parametrów na uzyskany wynik.

1.1 Dane do zadania

- poziom morza $h_m = 6378137\,m$
- wysokość umieszczenia satelitów $h_s = 20000000\,m\,n.p.m.$
- prędkość sygnału z satelitów $v_s = 299792458\,m/s$
- dane o położeniu satelitów:

Nr	System	szerokość	długość
1	stopnie-min-sek	52°52'19.2"N	13°23'53.9"E
	stopnie dziesiętne	52.885907	13.395837
2	stopnie-min-sek	50°18'43.4"N	12°22'24.1"E
	stopnie dziesiętne	50.312052	12.373351
3	stopnie-min-sek	47°47'48.9"N	19°22'54.7"E
	stopnie dziesiętne	47.796902	19.381854
4	stopnie-min-sek	50°37'10.5"N	26°14'39.3"E
	stopnie dziesiętne	50.619584	26.244260
5	stopnie-min-sek	55°29'17.8"N	28°47'15.1"E
	stopnie dziesiętne	55.488272	28.787526

- czasy dotarcia sygnału od poszczególnych satelitów:

Nr. satelity	czas nadejścia sygnału [s]
1	6.681975504357193e-02
2	6.684736756249902e-02
3	6.676997507374220e-02
4	6.675806218267925e-02
5	6.687535792178521e-02

1.2 Przejście na współrzędne kartezjańskie

Za początek układu współrzędnych kartezjańskich przyjęto środek Ziemi. Wzory opisujące przejście z układu sferycznego do kartezjańskiego wyglądają następująco:

$$\begin{cases} x = rcos\theta cos\phi \\ y = rcos\theta sin\phi \\ z = rsin\theta \end{cases}$$

gdzie r oznacza promień wodzący, θ szerokość geograficzną a ϕ długość geograficzną.

2 Rozwiązanie

2.1 Sformułowanie układu równań opisującego szukane położenie w układzie współrzędnych kartezjańskich

Równanie określające odległość od satelity do nas można zapisać na dwa sposoby. Z jednej strony mamy dystans od podanych współrzędnych satelitów:

$$s = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2},$$

gdzie i oznacza indeks satelity (od 1 do 5), a poszukiwane współrzędne punktu na kuli Ziemskiej to x, y, z.

Z drugiej możemy go wyznaczyć na podstawie czasu propagacji sygnału i prędkości z jaką dociera do szukanego punktu:

$$s = v_s t_i$$

W ten sposób otrzymano układ równań, na podstawie którego można sformułować zadanie optymalizacji.

2.2 Sformułowanie zadania optymalizacji bez ograniczeń stosując metodę najmniejszych kwadratów

Mając dwie postacie wzorów na odległość s, można obie podnieść do kwadratu i odjąć stronami uzyskując:

$$0 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - (v_s t_i)^2$$

Wykorzystując metodę najmniejszych kwadratów poszukiwane jest minimum funkcji kwadratowej:

$$\min_{x,y,z} f(x, y, z) = \sum_{i=1}^{5} ((x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - (v_s t_i)^2)^2$$

2.3 Wyznaczenie współrzędnych położenia

(a) Za pomocą programu Matlab

Wyznaczenie położenia odbyło się przy pomocy funkcji `lsqnonlin` z *Optimalization Toolbox*. Korzystając z metody *Levenberg'a - Marquartdt'a* uzyskano współrzędne X, Y, Z szukanego położenia:

- X = 3719945.14681953 m
- Y = 1439160.63345815 m
- Z = 4977343.05407747 m

Kod z programem:

```
clear;

hm = 6378137;
hs = 20000000;
vs = 299792458;

r = hm+hs;
theta = [52.885907 50.312052 47.796902 50.619584 55.488272];
fi = [13.395837 12.373351 19.381854 26.244260 28.787526];

global xi yi zi;

xi = r.*cos(deg2rad(theta)).*cos(deg2rad(fi));
yi = r.*cos(deg2rad(theta)).*sin(deg2rad(fi));
zi = r.*sin(deg2rad(theta));

ti = [6.681975504357193e-02 6.684736756249902e-02 6.676997507374220e-02 ...
        6.675806218267925e-02 6.687535792178521e-02];

global vsti

vsti = vs .* ti;

x0 = [0 0 0];

OPTIONS = optimset('Algorithm', 'levenberg-marquardt', 'Jacobian', 'on');

[P, resnorm, residual, exitflag, output, lambda, Jacobian] = lsqnonlin(@FUN, x0, ...
                                                                []; [], OPTIONS);

x = P(1);
y = P(2);
z = P(3);
rn = sqrt(x^2+y^2+z^2);
fin = rad2deg(atan(y/x));
thetan = rad2deg(asin(z/rn));


function [k, J] = FUN(x)
    global xi yi zi vsti;

    k = (x(1) - xi).^2 + (x(2) - yi).^2 + (x(3) - zi).^2 - vsti.^2;
    if nargout > 1
        J = [2*x(1)-2.*xi; 2*x(2)-2.*yi; 2*x(3)-2.*zi].';
    end

end
```

(b) Za pomocą solvera MINOS we współpracy z AMPL

Do skorzystania z solvera MINOS do rozwiązania zadania optymalizacji, posłużono się serwerem NEOS. Przyjmuje on trzy pliki w jezyku AMPL - z modelem, danymi oraz komendami do wywołania. Potrzebne dane do solvera, czyli w tym przypadku współrzędne kartezjańskie satelitów oraz droga wyliczona z użyciem wzoru $s = v * t$, zostały uprzednio obliczone i skopiowane z Matlaba. Otrzymane współrzędne są bardzo bliskie tym z poprzedniego podpunktu ale nie identyczne:

- X = 3719950 m
- Y = 1439160 m

- Z = 4977340 m

Widać, że wartości są zaokrąglone. Wartości parametrów wymagały przeskalowania, żeby otrzymać prawidłowe rozwiązanie. Kod w AMPL:

- `model.mod`:

```
set SATELITY;
param r := 6378137;
param x{SATELITY} >0;
param y{SATELITY} >0;
param z{SATELITY} >0;
param w := 0.00001;

param vst{SATELITY} >0;

var X >= -r, <=r;
var Y >= -r, <=r;
var Z >= -r, <=r;

minimize f:
    sum{i in SATELITY} (w*(X-x[i])^2+w*(Y-y[i])^2+w*(Z-z[i])^2-w*(vst[i])^2)^2 ;
```

- `dane.dat`:

```
set SATELITY := s1 s2 s3 s4 s5;

param: x y z vst :=
s1 15483627.7584050 3687531.60620285 21034863.7666892 20032058.6074703
s2 16453955.2570464 3609614.61613039 20298870.6659484 20040336.6323911
s3 16715574.7203245 5880533.27785464 19540087.0450482 20017134.9479559
s4 15010839.8760186 7400657.58739250 20388993.7241020 20013563.5530623
s5 13098137.7195915 7197049.11651869 21735854.7999337 20048727.9310018 ;
```

- `cmd.run`:

```
solve;
display X;
display Y;
display Z;
```

3 Testy wpływu zmian parametrów zadania na uzyskiwane wyniki

Po wyznaczeniu rozwiązania sprawdzono wpływ zmian wartości punktu startowego, dokładności w teście STOPu metody oraz zaburzeń w danych na analizowane zadanie.

3.1 Wpływ zmian punktu startowego

Nr.	punkt startowy x_0, y_0, z_0	`resnorm` - kwadratowa norma residuum
1	0, 0, 0	3.8642e+03
2	3e+06, 3e+06, 3e+06	0.0117
3	3.7199e+06, 1.4392e+06, 4.9773e+06	0.0078
4	3.7199e+06, -1.4392e+06, 4.9773e+06	0.0312
5	3.7199e+06, -1.4392e+06, -4.9773e+06	3.8888e+09
6	1.8599e+06, 0.7196e+06, 2.4886e+06	0.0156
7	0, 0, 6378137	0.3203
8	3.72e+07, 1.44e+07, 4.98e+07	1.7551e+25

Analizę przeprowadzono sprawdzając, dla różnych punktów startowych, wartość parametru `resnorm`. Jest to podniesiona do kwadratu norma residuum, zwracana przez funkcję `lsqnonlin` po zakończeniu optymalizacji. Dla środka układu współrzędnych kartezjańskich - (0, 0, 0), widać, że błąd jest dość duży. W próbie nr. 2 - punkt w środku Ziemi - wynik jest znacznie dokładniejszy. W próbie nr. 3 sprawdzono dokładność dla punktu startowego znajdującego się bardzo blisko szukanego rozwiązania. W tym przypadku uzyskana dokładność jest największą ze wszystkich uzyskanych na drodze testów. W kolejnych próbach pokazano, jak może zmieniać się wartość `resnorm` w zależności od lokalizacji na powierzchni Ziemi punktu startowego. W próbach nr. 4 i 5 rozważono różne ćwiartki kuli Ziemskiej. Próba nr. 6 to punkt w środku Ziemi, gdzie wartości współrzędnych są o połowę mniejsze od tych z rozwiązania. W ostatniej, nr. 8, przyjęto punkt startowy całkowicie poza kulą Ziemską i tutaj też błąd przybliżenia był największy.

3.2 Wpływ dokładności w teście STOPu

W kolejnym kroku zmierzono wpływ dokładności STOPu metody. Optymalizacja zostaje zakończona jeśli przekroczona zostaje tolerancja w wartościach funkcji (`TolFun`) lub wielkości kroku (`TolX`). Oba parametry w domyśle przyjmują wartość 1e-06. Dla punktu startowego (0, 0, 0) przedstawiono analizę w postaci poniższej tabelki:

`TolFun`	`TolX`	`resnorm`	liczba iteracji
1e-06	1e-06	3.8642e+03	4
1e-8	1e-8	3.8642e+03	4
1e-10	1e-10	0.0430	5
1e-12	1e-12	0.0430	5
1e-14	1e-14	0.0430	6
1e-16	1e-16	0.0039	8
1e-40	1e-40	0.0039	8

Zmniejszając wartości obu tolerancji zauważono dwie zależności. Dokładność rozwiązania jest coraz większa aż do osiągnięcia pewnej wartości maksymalnej, przy której mimo znacznego zmniejszania tolerancji, błąd przybliżenia pozostaje stały i niezerowy. W miarę kolejnych prób zmniejszania tolerancji, rosła także liczba potrzebnych iteracji algorytmu optymalizacji.

3.3 Wpływ zaburzeń w danych

W celu zmierzenia wpływu zaburzeń w danych na uzyskiwane wyniki, postanowiono przeanalizować działanie optymalizacji dla coraz mniej precyzyjnych wartości czasów t_i dotarcia sygnału z satelity, ucinając kolejne cyfry po przecinku.

Liczba uciętych miejsc po przecinku	`resnorm`
0	3.8642e+03
1	3.8645e+03
2	3.9261e+03
5	2.1961e+06
8	4.6371e+13
12	9.0277e+20

Jak widać, już nawet pojedyncze usunięcie wartości po przecinku, sprawia że błąd przybliżenia rośnie. W miarę usuwania kolejnych wartości staje się on coraz większy.

4 Znalezione położenie w Google Maps

Wyznaczone wartości położenia w układzie kartezjańskim następnie przeliczono z powrotem do układu "geograficznego", za pomocą następujących wzorów:

$$\begin{cases} r_n = \sqrt{X^2 + Y^2 + Z^2} \\ \theta_n = \arcsin\left(Z/r_n\right) \\ \phi_n = \arctan\left(Y/X\right) \end{cases}$$

Dzięki temu uzyskano:

$$\begin{cases} r_n = 6378332\,m = 199\,m\,n.p.m \\ \theta_n = 51.2928\,N = 51°17'34"\,N \\ \phi_n = 21.1503\,E = 21°09'01"\,E \end{cases}$$

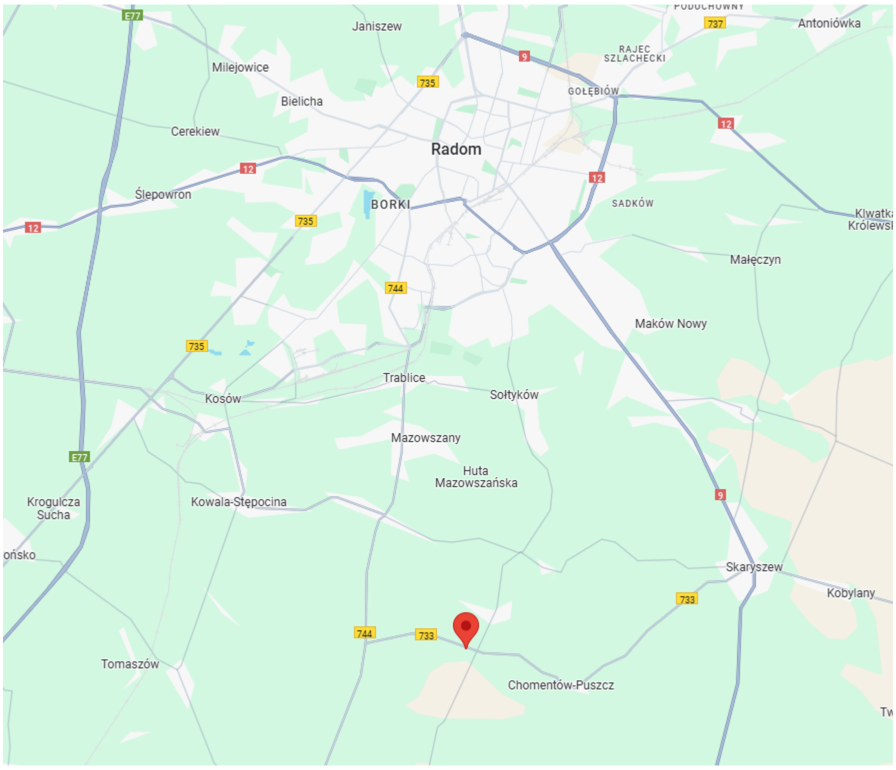

Figure 1: Zrzut ekranu - Google Maps, wyznaczona lokalizacja

Figure 2: Zrzut ekranu - Google Maps, wyznaczona lokalizacja

Figure 3: Zrzut ekranu - Google Earth, wyznaczona lokalizacja

5 Wnioski

W powyższym sprawozdaniu zostało przedstawione sformułowanie i rozwiązanie zadania optymalizacji bez ograniczeń. Celem było wyznaczenie pozycji odbiorcy na podstawie danych o położeniu satelitów i czasów dotarcia od nich sygnałów. Do zapisania zadania wykorzystano metodę najmniejszych kwadratów. Rozwiązanie oparto o dwa różne podejścia - metodę *levenberg'a-marquartdt'a* w *lsqnonlin* w Matlabie oraz solvera MINOS przy pomocy AMPL. W obu przypadkach uzyskano, w przybliżeniu, to samo rozwiązanie. W kolejnym kroku przeprowadzono analizę wpływu różnych parametrów zadania optymalizacji na uzyskiwane wyniki poprzez porównywanie błędu przybliżenia. Zauważono, że najwyższą dokładność uzyskuje się przy wybraniu punktu startowego w miarę blisko przewidywanego rozwiązania, a także gdy ustawi się wystarczająco małą tolerancję zmian w wartości funkcji oraz kroku w kryterium STOPu. Im większa dokładność danych wejściowych (mniejsze zaburzenia) tym również mniejszy błąd przybliżenia.